**En cada seguimiento, se deberá elaborar y refinar el documento de arquitectura, tal y como se muestra a continuación.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ingeniería de Sistemas II** | | | **Ingeniería de Sistemas III** |
| Versión No. 1  ***Seguimiento No. 1*** | Versión No. 2  ***Seguimiento No. 2*** | Versión No. 3 y No. 4  ***Seguimiento No. 3 y No. 4*** |  |
| Introducción  Metas y limitaciones   * Plataforma técnica * Descripción de la portabilidad Confiabilidad/Disponibilidad * Desempeño   Vistas de escenarios  Vista Lógica  Vista de implementación  Vista Física  Estándares de Usabilidad  Estándares de la Base de Datos  Estándares de Programación  Estándares de Documentación | Refinar la versión 1  Elaborar:  Metas y limitaciones  Portabilidad  Seguridad  Vistas por procesos  Definición de Estándares | Al finalizar cada Sprint se deberá refinar la versión anterior del documento de arquitectura. | Al finalizar cada Sprint se deberá refinar la versión anterior del documento de arquitectura.  Elaborar:   * Guía de instalación del sistema * Procedimiento para el mantenimiento del sistema. |

**Nota:** esta primer hoja es solamente de referencia deben eliminarla cuando terminen de elaborar el primer seguimiento.

Los puntos que están en fondo blanco en la tabla anterior no se deben presentar en el seguimiento 1, por lo que solamente los puntos que tienen fondo amarillo serán los que deban entregar.

También es importante que dejen el documento intacto, es decir no le borren las secciones que no desarrollan, el mismo debe estar completo y se va desarrollando cada punto conforme se den los seguimientos.

# UNIVERSIDAD NACIONAL

**ESCUELA DE INFORMÁTICA**

**SEDE INTERUNIVERSITARIA DE ALAJUELA**

**INGENIERÍA DE SISTEMAS II**

# 

# Documento de la Arquitectura del Sistema

# (IS-008-V02)

**Sistema de Control de Parqueos (S.C.P.)**

**Documento de Arquitectura de Software**

**Versión 1.0**

**Historial de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 23 de agosto de 2017 | 1.0 | Estándares de diseño, programación y bases de datos.  Riesgos  PAP | Alex José Baltodano Paniagua  José Carlos Barrantes Araya  Andrés Alberto Barrantes Murillo  Greivin Barrantes Segura  Josué Castro Villalobos |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introducción

## Propósito

El propósito de este documento es resumir la arquitectura del sistema, así como presentar el manual técnico del sistema.

## Alcance

Este documento corresponde al proyecto de software desarrollado utilizando Scrum, el cual se denomina Sistema de Control de Parqueos implementado en la Empresa Llobet.

Con relación a la arquitectura del software cubriendo los siguientes aspectos:

**Vista de Escenarios:** Esta vista va a ser representada por los casos de uso software y va a tener la función de unir y relacionar las otras vistas, esto quiere decir que desde un caso de uso podemos ver cómo se van ligando las otras vistas, con lo que tendremos una trazabilidad de componentes, clases, equipos, paquetes, etc., para realizar cada caso de uso. Para completar la documentación de esta vista se pueden incluir el diagrama de casos de uso de UML.

**Vista lógica:** En esta vista se representa la funcionalidad que el sistema proporcionara a los usuarios finales. Es decir, se ha de representar lo que el sistema debe hacer, y las funciones y servicios que ofrece. Esta vista presenta el modelo de tres capas: Interfaz, negocio y datos.

**Vista de implementación:** Conocida como vista de desarrollo. Esta vista se ocupa de la gestión del software; se va a mostrar cómo está dividido el sistema software en componentes y las dependencias que hay entre esos componentes. Para completar la documentación de esta vista se pueden incluir los diagramas de componentes, de paquetes, y de despliegue.

**Vista física:** La vista física se centra en los requisitos no funcionales, tales como la disponibilidad del sistema, la fiabilidad (tolerancia a fallos), ejecución y escalabilidad. Y también presenta cómo los procesos, objetos, etc., corresponden a nodos de proceso: Componentes ( nodos de proceso), Conectores( LAN, WAN, bus), Contenedores (subsistemas físico).

**Vista de procesos:** Esta vista también especifica que hilo de control ejecuta cada operación identificada en cada clase identificada en la vista lógica. La vista se centra por tanto en la concurrencia y distribución de procesos.

## Documentos relacionados.

|  |  |  |
| --- | --- | --- |
| **Código** | **Descripción** | **Anexo** |
|  |  |  |
|  |  |  |
|  |  |  |

# Metas y limitaciones

En este apartado se presentan los requerimientos técnicos y condiciones especiales que fueron consideradas en el desarrollo e implementación del Sistema. Se tratan los requisitos no funcionales relacionados con la ejecución, disponibilidad, tolerancia a fallos, integridad, etc.

## Plataforma técnica

*[Conjunto de plataformas y tecnologías en las cuales correrá el producto terminado]*

## Portabilidad

*[Describir la capacidad de que posee un software para ejecutarse en diferentes plataformas. En este apartado debe describirse una guía de instalación del sistema]*

## Seguridad

*[Describir los mecanismos de seguridad que manejará el software. En esta sección debe describirse la seguridad a nivel de autentificación de usuarios, composición del password, bitácoras de transacciones.]*

#### Registro de usuarios nuevos

*[En este apartado se debe describir la forma en que el Administrador del Sistema agrega y da de baja a los Usuarios, así como la asignación de privilegios en los componentes de software]*

#### Control de acceso de los usuarios y los componentes del sistema.

*[Se debe describir los perfiles de los usuarios y los accesos que tienen en el sistema, por ejemplo:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Perfil de usuario* | *Consulta de cliente* | *Registro de clientes* | *Registro de pagos* | *Cierre de Caja* | *Eliminar depósitos* | *Generar asientos* |
| *Cajero* | *x* |  | *x* | *x* |  |  |
| *Plataformista* | *x* | *x* |  |  |  |  |
| *Jefe de Cajeros* | *x* | *x* |  | *x* | *x* |  |
| *Auxiliar Contable* | *x* |  |  |  |  | *x* |

*]*

#### Bitácoras de transacciones

*[Las organizaciones requieren llevar un control interno sobre las transacciones de mayor riesgo, por ejemplo al eliminar un depósito, hay dinero de por medio y debe justificarse. Es posible que esta transacción esté limitada a un funcionario de mayor rango y se requiera que el sistema genere una bitácora con la fecha, el usuario que hizo la transacción y una justificación]*

#### Controles sobre la composición y asignación del password

*[La seguridad del sistema para asignar password debe definir aspectos como cantidad de caracteres, composición, cada cuánto debe renovarse el password, cómo cambiar la contraseña, entre otros. Este apartado debe describir cómo se acordó con el negocio estos aspectos y cómo quedó definido en el sistema.]*

## Persistencia

Entre los diferentes mecanismos que utilizaremos para la persistencia de los datos están la serialización de los datos y acceso directo a la base de datos.

## Confiabilidad/Disponibilidad

*[Describir el nivel de exigencia en términos de disponibilidad y como la arquitectura los garantiza]*

## Desempeño

*[Describir los requisitos de desempeño del software, tales como duración por transacción, cantidad de usuarios concurrentes soportados, entre otros]*

#### Requisitos de desempeño

#### Volumen transaccional

*[En esta sección se debe detallar la cantidad de transacciones esperadas durante un periodo, por ejemplo mensual. Este volumen permitirá tomar decisiones en la arquitectura, tales como tamaño del disco, velocidad de procesamiento, diseño de clases, entre otras. La estimación debe realizarse con base en las operaciones del negocio, por ejemplo: -Cantidad de clientes, -Cantidad de pedidos, Cantidad de pagos, Cantidad de productos actuales y su incremento mensual]*

# Representación de la arquitectura

*[En esta sección se describe cual es la arquitectura de software para el sistema y como se representa. Su organización por capas, o bien patrones de arquitectura utilizados (SOA, BPM, etc. Se debe de especificar que en orden de detallar el software de la manera más precisa posible, la estructura del documento sigue el modelo “4+1” vistas de Kruchten]*

## Vista de Escenarios

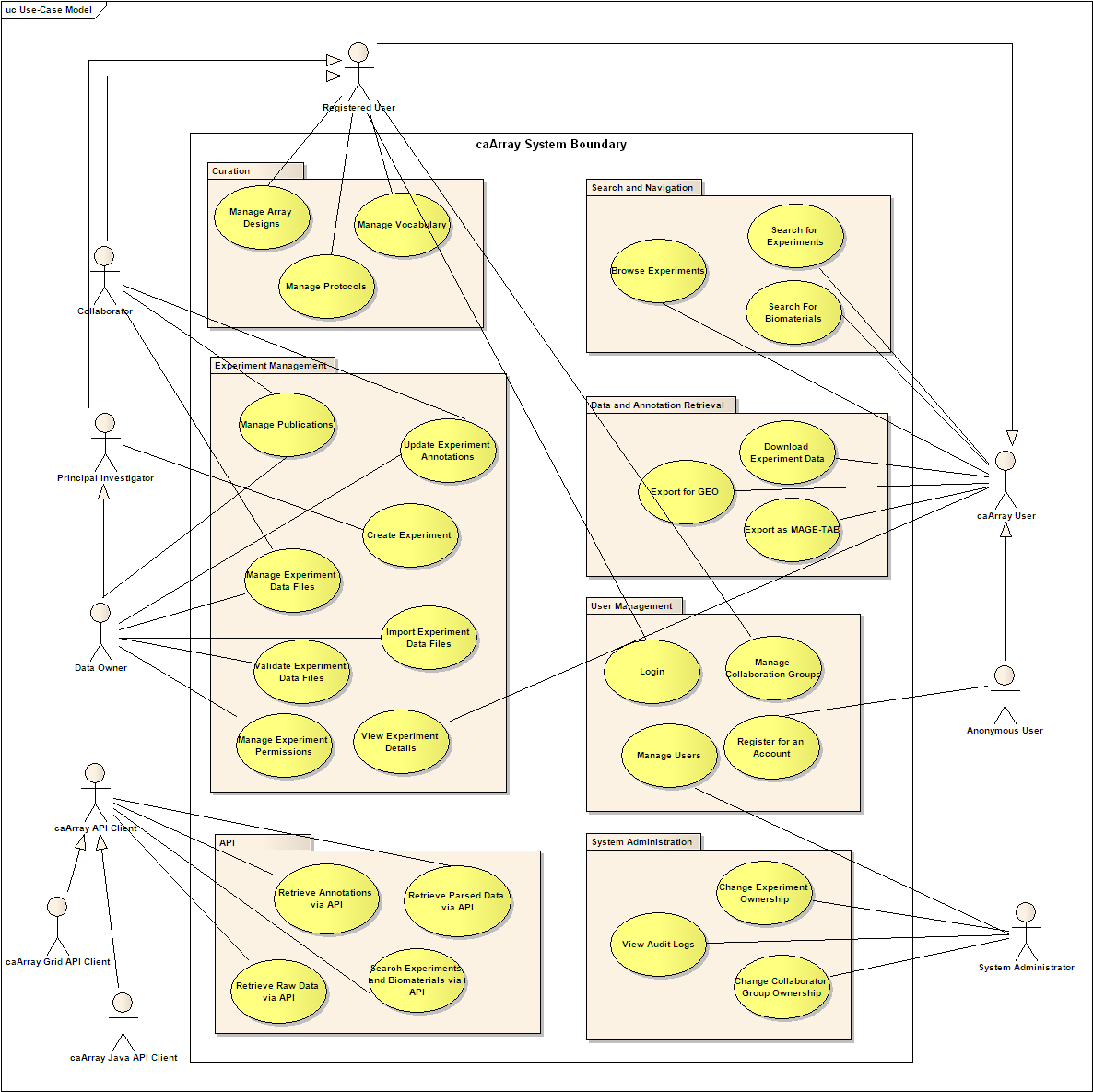
La vista de escenarios es usada para vincular las otras vistas, es una vista integradora de la funcionalidad a la que responde la arquitectura para dar trazabilidad a los componentes; es un punto de partida desde la perspectiva del usuario final donde se plasma la visión que estos tienen del negocio y de la aplicación para así describir el modelo de negocio que la arquitectura debe cubrir.

Esta vista estará descrita en términos de Casos de Uso y muestra los Subsistemas y Módulos en los que se divide la aplicación y la funcionalidad que brinda dentro de cada uno de ellos.

### Diagrama de Casos de Uso

*[Esta sección se debe copiar el diagrama de casos de uso. Se recomienda subdividir los casos de uso según la estructura de subsistemas del software tal y como se muestra en la figura 1]*

**Figura 1.** Vista de Escenarios: CU de los módulos del sistema



***Fuente: US National Institute of Health***

### Descomposición en sub-sistemas y módulos

*[Esta sección describe los distintos subsistemas en los que se divide el software. Además de describir el contenido y funcionalidad de cada subsistema o módulo, se debe representar mediante un diagrama de paquetes de UML como se muestra en la figura 3]*

**Subsistema A:**

*[Detallar el contenido y funcionalidad del subsistema A]*

.

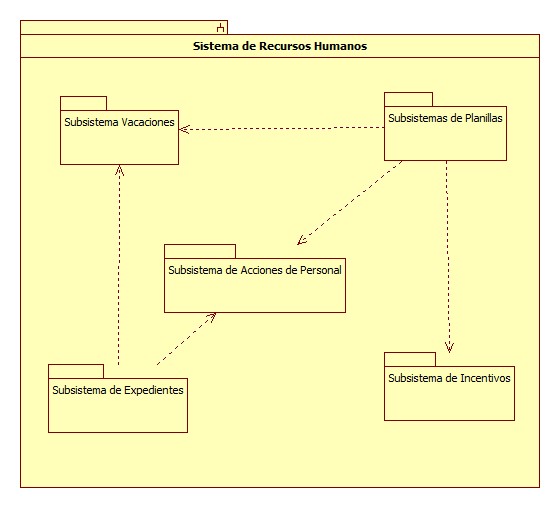
.

.

**Subsistema N:**

*[Detallar el contenido y funcionalidad del subsistema N]*

**Figura 2.** Descomposición en Subsistemas o Módulos



***Fuente: elaboración propia***

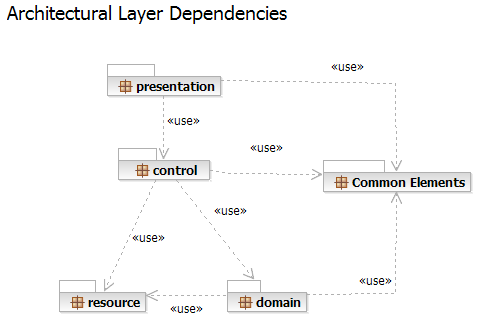
## Vista lógica

Esta sección describe las partes de la arquitectura más significativas del modelo de diseño desde la perspectiva de diseño del desarrollador, tal como su descomposición en paquetes y subsistemas, y su subsecuente descomposición en clases.

### Descomposición en Capas

*[En esta sección se deben de representar las distintas capas en las que es desarrollado el software así como las relaciones entre estas. Para realizar esta representación se puede utilizar un diagrama de componentes en UML donde cada componente presenta una capa. Se debe detallar el contenido y la funciona de cada capa.]*

**Figura 3.** Dependencias entre capas

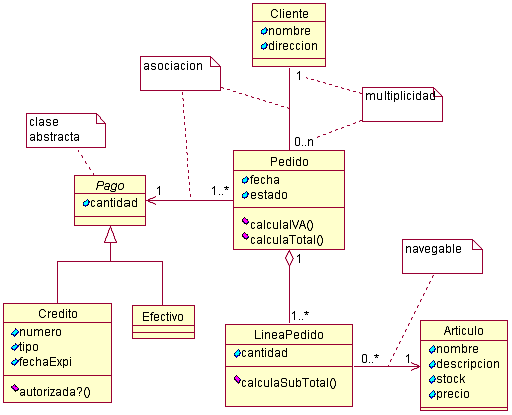


***Fuente: www.cse.hcmut.edu.vn***

### Capa de interfaz (presentación).

*[Esta sección se presenta el diagrama general de clases diseño del sistema como se muestra en la siguiente figura]*

**Figura 4.** Diagrama general de clase



***Fuente: elaboración propia***

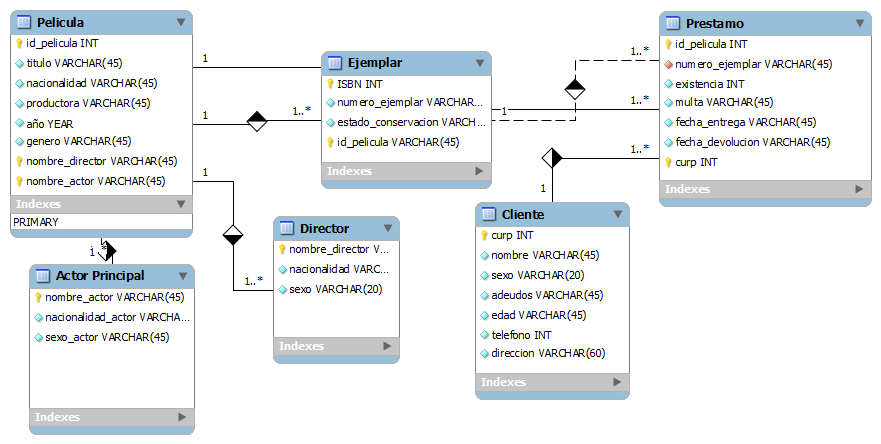
### Capa de negocio (Control).

*{Es la encargada de administrar los accesos a la capa de dominio. La capa de dominio contiene todos los componentes relacionados con la lógica de negocio. Reúne todos los subsistemas que satisfacen las necesidades de un dominio de negocio particular. También contiene el modelo de objetos de negocio}*

### Capa de datos.

*[Describe las relaciones entre las entidades de persistencia del software representadas mediante un modelo relacional del sistema, o bien un modelo de entidades de objetivos en caso de que se utilicen bases de datos orientados a objetos]*

**Figura 5.** Ejemplo de Modelo Relacional



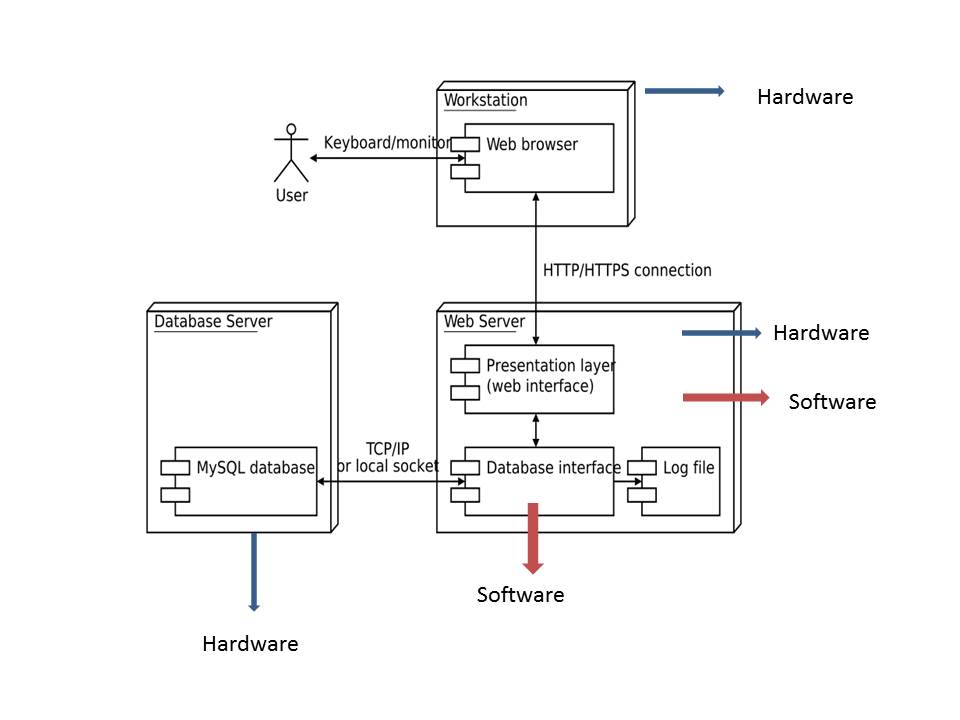
***Fuente: http://basesdedatosbydante.blogspot.com/***

## Vista por procesos

## Vista de implementación

*[Muestra los componentes de software que se planean utilizar para la implementación del sistema, a nivel de software y de hardware; y la manera en que estos estarán relacionados]*

**Figura 6.** Diagrama de componentes

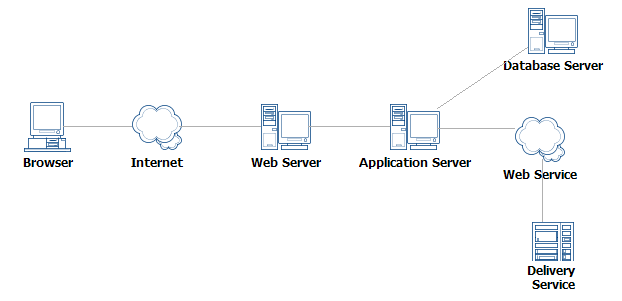


***Fuente: elaboración propia***

## Vista Física

*[Muestra los componentes de infraestructura física involucrados en el funcionamiento del software, tal y como se aprecia en la figura 7. En esta vista indican si el sistema es web enable o cliente servidor, por ejemplo.]*

**Figura 7.** Diagrama de componentes físicos



***Fuente: www.cse.hcmut.edu.vn***

# Definición de estándares

## Estándares de Usabilidad

*[Se debe detallar cómo están definidos los estándares en la empresa o bien cómo lo definieron. Hay organizaciones que cuentan con un “Registro de Marca” que incluye los logos, los colores, los tamaños de las letras y la distribución de los espacios en las pantallas. En caso de que la empresa no cuente con un estándar, el equipo desarrollador en conjunto con la Empresa, deben definir estos estándares que además deben cumplir con las Heurísticas de Nielsen]*

### Descripción de la forma en que acordaron la usabilidad del sistema

### Implementación de Heurísticas de Nielsen.

| Heurística | Definición (Responde al cómo se va a implementar en el sistema) |
| --- | --- |
| Estándares de usabilidad | Colores, marcas, tipos de letra |
| Ayuda en línea |  |
| Prevención de errores |  |
| .. |  |

## Estándares de la Base de Datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código** | **Descripción** | **Aplicación** | **Detalle del estándar** | **Ejemplo** |
| EBD01 | Nombre de las Tablas | Base de Datos | Los nombres que se usarán en las tablas seguirán un formato de palabra en mayúscula completamente y en plural. | CREATE TABLE USUARIOS; |
| EBD02 | Separación de palabras por nombre de tablas | Base de Datos | |  |  |  | | --- | --- | --- | | **N** | **-** | **M** | | XX | - | XX |   Detalle componente “\_“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Guión separador | \_ |   Detalle componente “N“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Primer palabra del nombre de la tabla | Caracteres del nombre |   Detalle componente “M“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Segunda palabra del nombre la tabla | Caracteres del nombre | | SELECT \* FROM CARROS\_ACTUALES; |
| EBD03 | Formato de la fecha | Bases de datos | Formato Fecha   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **dd** | **/** | **mm** | **/** | **yyyy** | | XX | / | XX | / | XXXX |   Detalle componente “dd“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Día de la fecha | Número de día |   Detalle componente “/“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Separador de la fecha | / |   Detalle componente “mm“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Mes de la fecha | Número de mes |   Detalle componente “yyyy“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Año de la fecha | Número de año | | 22/08/2017 |
| EBD04 | Formato de la hora | Bases de Datos | Formato hora   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **HH** | **:** | **mm** | **:** | **ss** | | XX | : | XX | : | XXXX |   Detalle componente “HH“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Hora | Hora en formato de 24 horas |   Detalle componente “:“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Separador de la hora | : |   Detalle componente “mm“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Minutos | Tiempo en minutos |   Detalle componente “ss“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Segundos | Tiempo en segundos | | 14:20:54 |
| EBD05 | Nombre de los Atributos | Bases de Datos | |  |  |  | | --- | --- | --- | | **n** | **\_** | **M** | | xxxx… max 10  Caract… xxxx | \_ | Xxxx…max 10 caract…Xxxx  Primer letra en mayúscula |   Detalle componente “n“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Primer palabra del nombre del atributo | Caracteres del nombre |   Detalle componente “\_“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Separador entre palabras del nombre | \_ |   Detalle componente “M“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Segunda palabra del nombre del atributo | Caracteres del nombre | | num\_Cedula |
| EBD06 | Id’s de tablas | Bases de Datos | Cada tabla va a tener su propio id autoincrementable, para que los otros atributos de la tabla no tengan referencia a otras llaves primarias o foráneas. | Usuarios   |  |  | | --- | --- | | **Tipo** | **Nombre** | | number | id | | varchar2(50) | nombre | | …. | …. | |

## Estándares de Programación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código** | **Descripción** | **Aplicación** | **Detalle del estándar** | **Ejemplo** |
| EP01 | Nombre de los Atributos | Programación | |  |  | | --- | --- | | **n** | **M** | | xxxx… max 10  Caract… xxxx | Xxxx…max 10 caract…Xxxx  Primer letra en mayúscula |   Detalle componente “n“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Primer palabra del nombre del atributo | Caracteres del nombre |   Detalle componente “M“   |  |  | | --- | --- | | **Tipo** | **Detalle** | | Segunda palabra del nombre del atributo | Caracteres del nombre | | numCedula |
| EP02 | Excepciones | Programación | Se seguirá un estándar de excepciones de errores con un mensaje detallado y registro. | Cantidad máxima de caracteres sobrepasada. |

## Estándares de Diseño

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código** | **Descripción** | **Aplicación** | **Detalle del estándar** | **Ejemplo** |
| ED01 | Marca de agua | Diseño | No se utilizarán marcas de agua en ninguna ventana ni en ningún reporte. |  |
| ED02 | Logo | Diseño | No se utilizará el logo en ninguna ventana, ningún reporte y tampoco en las facturas. |  |
| ED03 | Botones | Diseño | Se utilizará el formato de Oracle Forms 6i, no es fundamental agregar imágenes a los botones. |  |
| ED04 | Imágenes | Diseño | No es fundamental utilizar imágenes. |  |

## Estándares de Documentación

|  |  |
| --- | --- |
| **Código** | **Descripción** |
| ED-01 | Documentación los componentes |
| ED-02 | … |
| … |  |
|  |  |

# Guía de instalación del sistema

*<El sistema debe quedar instalado en la infraestructura de la empresa. Sin embargo, podría requerir instalarse nuevamente. En este apartado debe incluirse paso a paso la instalación de cada componente del sistema desde el sistema operativo, motor de base de datos, hasta incluir todos los componentes requeridos para que el sistema opere en una nueva plataforma.>*

# Procedimiento para el Mantenimiento del Sistema

*<Todo Sistema requiere de mantenimiento, este comprende las mejoras, atención a los atención de incidentes, respaldo y recuperación de la base de datos. Antes de finalizar el proyecto debe sugerirse a la Empresa un conjunto de pasos para atender los incidentes y eventuales mejoras al Sistema de Información. En el mantenimiento del sistema participan los usuarios y un desarrollador, las facultades técnicas de cada parte deben estar descritas en este procedimiento, así como las responsabilidad de cada participante y pasos a seguir.>*

## Introducción

## Roles y Responsabilidades

## Descripción del Proceso

## Diagrama de flujo

## Anexos