**Diseño**

**Documento de Diseño(DDIS\_V1.0\_2016)**

**Área (REQM)**

Sistema de Reserva de Asientos

Diseño

22/06/2016

BarrioKing

1. **Introducción**

El presente documento describe el diseño externo del Sistema de Reserva de Asientos. Este diseño incluye el diseño de interfaces gráficas (prototipos), diseño de procedimientos (rutinas que no tienen interfaz gráfica), diseño de reportes que puedan complementar el funcionamiento y objetivo de cada diseño. Cada diseño de presente documento tiene una descripción detallada de lo que pretende implementar

El diseño tiene como objetivo general ser una guía detallada de cómo se van a implementar los requisitos del sistema y de cómo el usuario final lo va a percibir sin entrar a detalles técnicos de implementación. En otras palabras, describe la funcionalidad esperada por el usuario final.

1. **Descripción general del sistema**

El sistema completo **considera tres usuarios: el usuario Empleado, el usuario Cliente y el usuario Administrador**. Y para cada uno de estos crear una aplicación acorde a las tareas que realizará cada uno, y los permisos que tenga.

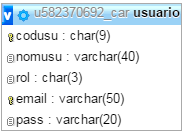
Para el usuario Cliente es necesaria una aplicación cliente que muestre los eventos activos y obtener una cuenta, el módulo de reserva mostrará los asientos disponibles y los que se pueden adquirir y efectuar la reserva. También recibirá mediante una notificación un mensaje en su correo.

La Sistema del administrador ofrece lo que es un CRUD de los eventos, el cual posee un desactivador automático cuando el evento ya haya concluido.

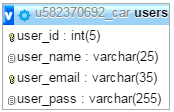
El APK del usuario empleado comprende la desencriptación de un código Qr que el cliente debe poseer al presentarse al evento

1. **Tablas de Base de datos**

* Tabla Usuario ADMIN



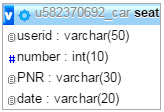
* Tabla Usuario Cliente



* Tabla Evento



* Tabla Asientos



**Script Base de Datos**

-- phpMyAdmin SQL Dump

-- version 4.5.1

-- http://www.phpmyadmin.net

--

-- Servidor: 127.0.0.1

-- Tiempo de generación: 20-06-2016 a las 06:38:31

-- Versión del servidor: 10.1.13-MariaDB

-- Versión de PHP: 5.6.20

--

-- Base de datos: `u582370692\_car`

--

--creacion de evento

CREATE EVENT actuali

ON SCHEDULE

EVERY 1 HOUR

DO

Update eventos set eve\_estado=’0’ where eve\_fecha<now();

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `eventos`

--

CREATE TABLE `eventos` (

`eve\_cod` char(5) COLLATE utf8\_bin NOT NULL,

`eve\_titulo` varchar(100) COLLATE utf8\_bin NOT NULL,

`eve\_lugar` varchar(100) COLLATE utf8\_bin NOT NULL,

`eve\_fecha` varchar(30) COLLATE utf8\_bin NOT NULL,

`eve\_estado` varchar(1) COLLATE utf8\_bin NOT NULL,

`eve\_cuerpo` mediumtext COLLATE utf8\_bin NOT NULL,

`eve\_foto1` varchar(50) COLLATE utf8\_bin NOT NULL,

`eve\_foto2` varchar(50) COLLATE utf8\_bin NOT NULL,

`eve\_foto3` varchar(50) COLLATE utf8\_bin NOT NULL,

`eve\_foto4` varchar(50) COLLATE utf8\_bin NOT NULL,

`eve\_creacion` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Volcado de datos para la tabla `eventos`

--

INSERT INTO `eventos` (`eve\_cod`, `eve\_titulo`, `eve\_lugar`, `eve\_fecha`, `eve\_estado`, `eve\_cuerpo`, `eve\_foto1`, `eve\_foto2`, `eve\_foto3`, `eve\_foto4`, `eve\_creacion`) VALUES

('ev001', 'Zaperoko en el Callao', 'llaoca', '2016-06-30 14:15:00', '1', '', 'ev001.jpg', 'ev001b.jpg', 'ev001c.jpg', 'ev001d.jpg', '2016-06-08 19:17:09'),

('ev002', 'DelCarajo', 'barranco', '2016-06-23 16:15:00', '1', '', 'ev002.jpg', 'ev002b.jpg', 'ev002c.jpg', 'ev002d.jpg', '2016-06-08 21:14:32'),

('ev003', 'peruvsbrazil', 'us', '2016-06-30 00:50:00', '1', '', 'ev003.jpg', 'ev003b.jpg', 'ev003c.jpg', 'ev003d.jpg', '2016-06-13 05:51:16');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `register`

--

CREATE TABLE `register` (

`userid` varchar(50) COLLATE utf8\_bin NOT NULL COMMENT 'userid',

`name` varchar(50) COLLATE utf8\_bin NOT NULL COMMENT 'name',

`email` varchar(50) COLLATE utf8\_bin NOT NULL COMMENT 'email',

`password` varchar(50) COLLATE utf8\_bin NOT NULL COMMENT 'password',

`address` varchar(500) COLLATE utf8\_bin NOT NULL COMMENT 'address',

`contact` varchar(50) COLLATE utf8\_bin NOT NULL COMMENT 'contact',

`message` varchar(500) COLLATE utf8\_bin NOT NULL COMMENT 'Message'

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `seat`

--

CREATE TABLE `seat` (

`userid` varchar(50) COLLATE utf8\_bin NOT NULL COMMENT 'userid',

`number` int(10) NOT NULL COMMENT 'seat number',

`PNR` varchar(30) COLLATE utf8\_bin NOT NULL COMMENT 'PNR',

`date` varchar(20) COLLATE utf8\_bin NOT NULL COMMENT 'date of booking'

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Volcado de datos para la tabla `seat`

--

INSERT INTO `seat` (`userid`, `number`, `PNR`, `date`) VALUES

('1', 43, '2016-06-23 16:15:00-43', '2016-06-23 16:15:00'),

('1', 44, '2016-06-23 16:15:00-44', '2016-06-23 16:15:00'),

('1', 45, '2016-06-23 16:15:00-45', '2016-06-23 16:15:00'),

('1', 1, '2016-06-23 16:15:00-1', '2016-06-23 16:15:00'),

('1', 2, '2016-06-23 16:15:00-2', '2016-06-23 16:15:00'),

('1', 1, '2016-06-23 16:15:00-1', '2016-06-23 16:15:00'),

('1', 2, '2016-06-23 16:15:00-2', '2016-06-23 16:15:00'),

('1', 3, '2016-06-23 16:15:00-3', '2016-06-23 16:15:00'),

('1', 1, '2016-06-23 16:15:00-1', '2016-06-23 16:15:00'),

('1', 2, '2016-06-23 16:15:00-2', '2016-06-23 16:15:00'),

('1', 3, '2016-06-23 16:15:00-3', '2016-06-23 16:15:00'),

('1', 4, '2016-06-23 16:15:00-4', '2016-06-23 16:15:00'),

('1', 5, '2016-06-23 16:15:00-5', '2016-06-23 16:15:00'),

('1', 6, '2016-06-23 16:15:00-6', '2016-06-23 16:15:00'),

('1', 46, '2016-06-23 16:15:00-46', '2016-06-23 16:15:00'),

('1', 47, '2016-06-23 16:15:00-47', '2016-06-23 16:15:00'),

('1', 48, '2016-06-23 16:15:00-48', '2016-06-23 16:15:00'),

('1', 46, '2016-06-23 16:15:00-46', '2016-06-23 16:15:00'),

('1', 47, '2016-06-23 16:15:00-47', '2016-06-23 16:15:00'),

('1', 48, '2016-06-23 16:15:00-48', '2016-06-23 16:15:00'),

('1', 7, '2016-06-23 16:15:00-7', '2016-06-23 16:15:00'),

('1', 8, '2016-06-23 16:15:00-8', '2016-06-23 16:15:00'),

('1', 9, '2016-06-23 16:15:00-9', '2016-06-23 16:15:00'),

('1', 9, '2016-06-23 16:15:00-9', '2016-06-23 16:15:00'),

('1', 10, '2016-06-23 16:15:00-10', '2016-06-23 16:15:00'),

('1', 10, '2016-06-23 16:15:00-10', '2016-06-23 16:15:00'),

('1', 10, '2016-06-23 16:15:00-10', '2016-06-23 16:15:00'),

('1', 10, '2016-06-23 16:15:00-10', '2016-06-23 16:15:00'),

('1', 10, '2016-06-23 16:15:00-10', '2016-06-23 16:15:00'),

('1', 11, '2016-06-23 16:15:00-11', '2016-06-23 16:15:00'),

('1', 11, '2016-06-23 16:15:00-11', '2016-06-23 16:15:00'),

('1', 12, '2016-06-23 16:15:00-12', '2016-06-23 16:15:00'),

('1', 40, '2016-06-23 16:15:00-40', '2016-06-23 16:15:00'),

('1', 39, '2016-06-23 16:15:00-39', '2016-06-23 16:15:00'),

('1', 1, '2016-06-30 14:15:00-1', '2016-06-30 14:15:00'),

('1', 2, '2016-06-30 14:15:00-2', '2016-06-30 14:15:00'),

('1', 41, '2016-06-23 16:15:00-41', '2016-06-23 16:15:00'),

('1', 42, '2016-06-23 16:15:00-42', '2016-06-23 16:15:00'),

('1', 37, '2016-06-23 16:15:00-37', '2016-06-23 16:15:00'),

('1', 38, '2016-06-23 16:15:00-38', '2016-06-23 16:15:00'),

('1', 37, '2016-06-23 16:15:00-37', '2016-06-23 16:15:00'),

('1', 38, '2016-06-23 16:15:00-38', '2016-06-23 16:15:00'),

('1', 1, '2016-06-30 00:50:00-1', '2016-06-30 00:50:00'),

('1', 2, '2016-06-30 00:50:00-2', '2016-06-30 00:50:00'),

('1', 3, '2016-06-30 00:50:00-3', '2016-06-30 00:50:00'),

('1', 4, '2016-06-30 00:50:00-4', '2016-06-30 00:50:00'),

('1', 5, '2016-06-30 00:50:00-5', '2016-06-30 00:50:00'),

('1', 6, '2016-06-30 00:50:00-6', '2016-06-30 00:50:00'),

('1', 7, '2016-06-30 00:50:00-7', '2016-06-30 00:50:00'),

('1', 14, '2016-06-30 00:50:00-14', '2016-06-30 00:50:00'),

('1', 21, '2016-06-30 00:50:00-21', '2016-06-30 00:50:00'),

('1', 28, '2016-06-30 00:50:00-28', '2016-06-30 00:50:00'),

('1', 35, '2016-06-30 00:50:00-35', '2016-06-30 00:50:00'),

('1', 42, '2016-06-30 00:50:00-42', '2016-06-30 00:50:00'),

('1', 34, '2016-06-23 16:15:00-34', '2016-06-23 16:15:00'),

('1', 35, '2016-06-23 16:15:00-35', '2016-06-23 16:15:00'),

('1', 36, '2016-06-23 16:15:00-36', '2016-06-23 16:15:00');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `slider`

--

CREATE TABLE `slider` (

`sli\_cod` char(5) NOT NULL,

`sli\_titulo` varchar(100) NOT NULL,

`sli\_foto` varchar(50) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `users`

--

CREATE TABLE `users` (

`user\_id` int(5) NOT NULL,

`user\_name` varchar(25) NOT NULL,

`user\_email` varchar(35) NOT NULL,

`user\_pass` varchar(255) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--

-- Volcado de datos para la tabla `users`

--

INSERT INTO `users` (`user\_id`, `user\_name`, `user\_email`, `user\_pass`) VALUES

(1, 'jeral', 'jeral@gmail.com', '202cb962ac59075b964b07152d234b70');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `usuario`

--

CREATE TABLE `usuario` (

`codusu` char(9) COLLATE utf8\_bin NOT NULL,

`nomusu` varchar(40) COLLATE utf8\_bin NOT NULL,

`rol` char(3) COLLATE utf8\_bin NOT NULL DEFAULT 'U',

`email` varchar(50) COLLATE utf8\_bin NOT NULL,

`pass` varchar(20) COLLATE utf8\_bin NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Volcado de datos para la tabla `usuario`

--

INSERT INTO `usuario` (`codusu`, `nomusu`, `rol`, `email`, `pass`) VALUES

('us0001', 'Administrador', 'A', 'carajo@admin.com', 'carajo123');

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `eventos`

--

ALTER TABLE `eventos`

ADD PRIMARY KEY (`eve\_cod`);

--

-- Indices de la tabla `slider`

--

ALTER TABLE `slider`

ADD PRIMARY KEY (`sli\_cod`);

--

-- Indices de la tabla `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`user\_id`),

ADD UNIQUE KEY `user\_email` (`user\_email`);

--

-- Indices de la tabla `usuario`

--

ALTER TABLE `usuario`

ADD PRIMARY KEY (`codusu`),

ADD UNIQUE KEY `email` (`email`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `users`

--

ALTER TABLE `users`

MODIFY `user\_id` int(5) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

DELIMITER $$

--

-- Procedimientos

--

CREATE PROCEDURE `borraEvent` (`cod` CHAR(5)) delete from eventos where eve\_cod=cod$$

CREATE PROCEDURE `borraSlider` (`cod` CHAR(5)) delete from slider where sli\_cod=cod$$

CREATE PROCEDURE `editaEvent` (`cod` CHAR(5), `titu` VARCHAR(100), `lugar` VARCHAR(100), `fecha` TIMESTAMP, `est` VARCHAR(1), `cuer` MEDIUMTEXT) update eventos set eve\_titulo=titu, eve\_lugar=lugar, eve\_fecha=fecha,eve\_estado=est, eve\_cuerpo=cuer where eve\_cod=cod$$

CREATE PROCEDURE `editaSlider` (`cod` VARCHAR(5), `tit` VARCHAR(100), `ext` VARCHAR(5)) BEGIN

IF isnull(ext) THEN

update slider set sli\_titulo=tit where sli\_cod=cod;

ELSE

set @foto:= concat(cod,'\_', TRUNCATE(rand() \*100 ,0 ),'.',ext);

update slider set sli\_titulo=tit, sli\_foto=@foto where sli\_cod=cod;

select @foto img;

END IF;

END$$

CREATE PROCEDURE `insertEvent` (`nom` VARCHAR(100), `lug` VARCHAR(100), `fec` TIMESTAMP, `est` VARCHAR(1), `des` MEDIUMTEXT) BEGIN

DECLARE co CHAR(5);

DECLARE foto varchar(50);

DECLARE foto2 varchar(50);

DECLARE foto3 varchar(50);

DECLARE foto4 varchar(50);

SET co = (select concat('ev',right(concat('000',right(IFNULL(max(eve\_cod),0),3)+1),3)) AS COD from eventos);

SET foto =(select concat('ev',right(concat('00',right(IFNULL(max(eve\_cod),'00'),2)+1),3),'.jpg') AS img from eventos);

SET foto2 =(select concat('ev',right(concat('00',right(IFNULL(max(eve\_cod),'00'),2)+1),3),'b.jpg') AS img from eventos);

SET foto3 =(select concat('ev',right(concat('00',right(IFNULL(max(eve\_cod),'00'),2)+1),3),'c.jpg') AS img from eventos);

SET foto4 =(select concat('ev',right(concat('00',right(IFNULL(max(eve\_cod),'00'),2)+1),3),'d.jpg') AS img from eventos);

insert into eventos (eve\_cod,eve\_titulo,eve\_lugar,eve\_fecha,eve\_estado,eve\_cuerpo,eve\_foto1,eve\_foto2,eve\_foto3,eve\_foto4) values(co,nom,lug,fec,est,des,foto,foto2,foto3,foto4);

select co,foto,foto2,foto3,foto4;

END$$

CREATE PROCEDURE `insertSli` (`nom` VARCHAR(200), `ext` VARCHAR(5)) BEGIN

DECLARE co CHAR(5);

DECLARE img varchar(15);

SET co = (select concat('sl',right(concat('000',right(IFNULL(max(sli\_cod),0),3)+1),3)) AS COD from slider);

SET img = (select concat(co,'.',ext));

insert into slider(sli\_cod,sli\_titulo,sli\_foto) values(co,nom,img);

select img;

END$$

CREATE PROCEDURE `mostEvent` () select \* from eventos where eve\_estado='1' ORDER BY RAND()$$

CREATE PROCEDURE `mostEventAct` () select \* from eventos where eve\_estado='1' ORDER BY RAND()$$

CREATE PROCEDURE `mostEventDetalle` (`cod` CHAR(5)) select \* from eventos where eve\_cod=cod$$

CREATE PROCEDURE `mostSlider` () select \* from slider$$

CREATE PROCEDURE `sp\_login` (`us` VARCHAR(50), `ps` VARCHAR(20)) BEGIN

declare resul varchar(100);

declare rola char(3);

set resul = (select nomusu from usuario where email=us and pass=ps);

set rola = (select rol from usuario where email=us and pass=ps);

IF (resul <> '') THEN

select 'success' res,resul,rola;

ELSE

select 'fail' res;

END IF;

END$$

DELIMITER ;

1. **Implementación de requisitos**

**Requisito de diseño:**

**Registro de clientes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Descripción de Funcionalidad** | | | | | |
| **Campos:** | | | | | |
|  | | User\_name | | Ingreso del nombre de los usuario | |
|  | | User\_email | | Ingreso del Mail del usuario que servirá de Login. | |
|  | | User\_pass | | Ingreso de la Clave del Usuario | |
| **Botones de Acción:** | | | | | |
|  | Registrar ahora | | | Graba los datos ingresados | |
| **2** | **Flujo de Eventos** | | | | | |
|  | Flujo Básico | | | | 1. El Cliente Ingresa sus Datos 2. El Cliente pulsa el botón Registrar 3. El Cliente espera la Alerta de Confirmación. 4. El Cliente ya posee una cuenta | |
|  | Flujo Alternativo | | | | * Si no se llenan todos los campos, se mostrará un mensaje de error. | |
| **3** | **Consideraciones y/o Validaciones** | | | | | |
| Reglas de  Negocio | | | * El Cliente debe ingresar todos los campos. * El Cliente grabará los datos, los cuales se guardarán en una base de datos. * Se generará un código automáticamente al guardar el trabajador. | | |
| Validaciones | | | Comprobar | | Comprobar que el Correo sea Legitimo |

**Proceso Reserva**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Descripción de Funcionalidad** | | | | | |
| **Campos:** | | | | | |
|  | | Usuario | | Ingreso del usuario creado | |
|  | | clave | | Ingreso de la Clave Establecida | |
|  | |  | |  | |
|  | |  | |  | |
| **Botones de Acción:** | | | | | |
|  | Iniciar | | | Inicia el Login | |
| **2** | **Flujo de Eventos** | | | | | |
|  | Flujo Básico | | | | 1. El Cliente ingresa al programa ingresando al login y colocar contraseña 2. El Cliente se dirige hacia la Lista de Eventos 3. El Cliente pulsa el botón Ver Detalle 4. El Cliente ingresa Reserva los Asiente Deseados y disponibles. 5. El Cliente pulsa el botón Confirmar 6. El Cliente espera el Mensaje de Confirmacion | |
|  | Flujo Alternativo | | | | * Si no se llenan todos los campos, se mostrará un mensaje de error. | |
| **3** | **Consideraciones y/o Validaciones** | | | | | |
| Reglas de  Negocio | | | * El Cliente debe ingresar todos los campos. * El Cliente grabará los datos, los cuales se guardarán en una base de datos. * Se generará un código automáticamente al guardar el articulo | | |
| Validaciones | | | Comprobar | | Comprobar que no se dejen espacios en blanco |

**Creación de Eventos Admin**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Descripción de Funcionalidad** | | | | | |
| **Campos:** | | | | | |
|  | | Usuario | | Ingreso del Usuario | |
|  | | Contraseña | | Ingreso de la contraseña | |
|  | |  | |  | |
|  | |  | |  | |
| **Botones de Acción:** | | | | | |
|  |  | Ingresar | | | Abre el formulario para registrar datos | |
| **2** | **Flujo de Eventos** | | | | | |
|  | Flujo Básico | | | | 1. El Admin ingresa al Sistema del Administrador 2. El Admin se dirige hacia el módulo de Eventos 3. El Admin pulsa el botón Registrar 4. El Admin Rellena los campos de Creación de Eventos | |
|  | Flujo Alternativo | | | | * Si no se llenan todos los campos, se mostrará un mensaje de error. | |
| **3** | **Consideraciones y/o Validaciones** | | | | | |
| Reglas de  La aplicación | | | * El Admin debe ingresar todos los campos. * El Admin grabará los datos, los cuales se guardarán en una base de datos. | | |
| Validaciones | | | Comprobar | | Comprobar que no se dejen espacios en blanco |

**Proceso Empleado**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Descripción de Funcionalidad** | | | | | |
| **Campos:** | | | | | |
|  | | Id\_evento | | Ingreso del Id del Evento | |
|  | | Id\_usuario | | Ingreso del Id del usuario | |
|  | | Id\_reserva | | Ingreso del Id de la Reserva | |
|  | |  | |  | |
| **Botones de Acción:** | | | | | |
|  |  | scanear | | | Scanea el código del Cliente | |
| **2** | **Flujo de Eventos** | | | | | |
|  | Flujo Básico | | | | 1. El Empleado scanea el Codigo QR 2. El Empleado Realiza la Consulta del Cliente 3. El Empleado lleva al Cliente a la Caja | |
|  | Flujo Alternativo | | | | * Si no se llenan todos los campos, se mostrará un mensaje de error. | |
| **3** | **Consideraciones y/o Validaciones** | | | | | |
| Reglas de  La aplicación | | | * El Empleado debe ingresar todos los campos. * El Empleado debe scanear el código en un lugar iluminado | | |
| Validaciones | | | Comprobar | | Comprobar que no se dejen espacios en blanco |

1. **Diagrama de Clases**

