	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017

1) Análisis y Diseño

Lea con atención y realice el análisis de la siguiente ficha de historia de usuario

Título: Gestionar reservaciones
Descripción <i>Como</i> administrador del hotel <i>Necesito que</i> el sistema me permita gestionar reservaciones para consultar la disponibilidad de las habitaciones del hotel y en caso tal generar una reservación. <i>De manera que</i> sea posible determinar en el momento necesario si hay disponibilidad para generar una reservación.
Criterio de aceptación Se debe poder observar que: <ol style="list-style-type: none"> 1) el sistema permite consultar las reservaciones existentes a partir de los criterios dados. 2) el sistema permite generar una reservación con la información requerida teniendo en cuenta la disponibilidad 3) el sistema no permite generar una reservación dado que no existe disponibilidad


A partir del análisis de la ficha de historia anterior:

- a. Identifique si la historia requiere incluir notas o aclaraciones adicionales

R: // Para generar una reserva se debe conocer el numero de la habitación y posteriormente sea de mayor facilidad la búsqueda de la habitación , para ello es necesario implementar “room” para conocer las habitaciones disponibles


- b. Realice el desglose del trabajo que sería necesario realizar para implementar esta historia.

1. Pagina web nueva
 - 1.1. Concepción
 - 1.1.1. Crear Diseño
 - 1.1.2. Planificar Contenido
 - 1.2. Implementación
 - 1.2.1. Crear contenido
 - 1.2.2. Actualizar contenido

	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017

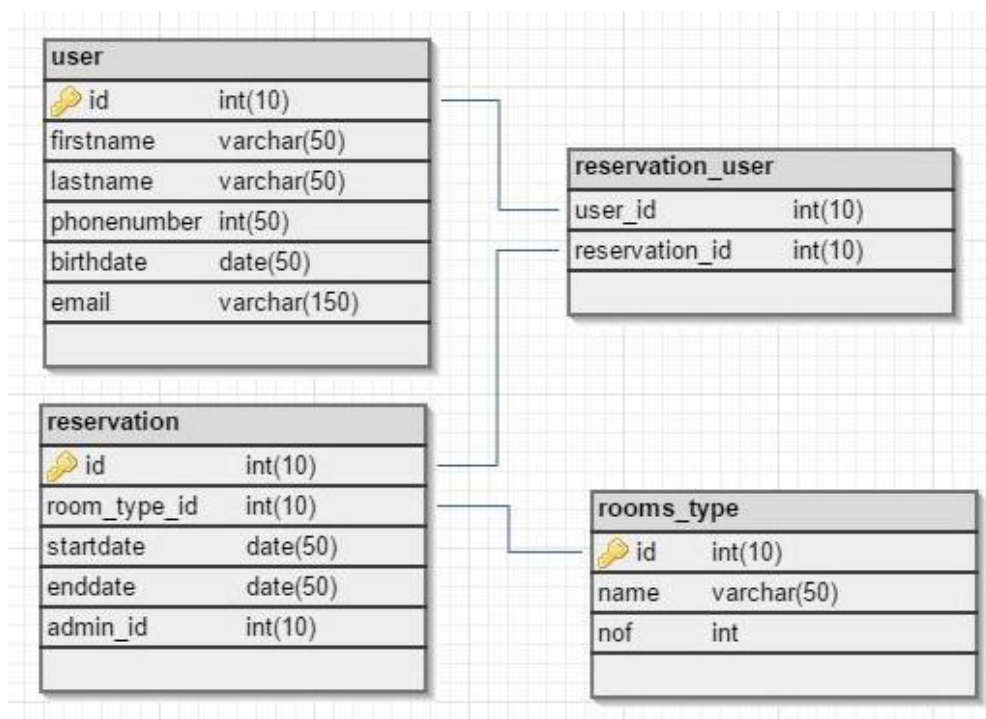
c. Escriba un caso de prueba para la historia

Propósito	Registrar servicio del cliente.
Prerrequisito	Usuario debe estar registrado o autenticado como administrador o recepcionista.
Datos correctos	<ul style="list-style-type: none"> • Id =201612971 • Nombre = Daniel • Apellido = piñeros • Telefono = 3118282951 • fecha de nacimiento : 1996 – 11 – 07 • Correo : Daniel.pineros01@uptc.edu.co
Datos incorrectos	<ul style="list-style-type: none"> • Id ="vacío" • Nombre = "vacío" • Apellido = "vacío" • Telefono = "vacío" • fecha de nacimiento : "vacío" • correo : "vacío"
Resultados esperados	<p>Si los datos fueron enviados se envía una notificación con los datos correctos</p> <p>Si los datos fueron enviados incorrectamente se envía una notificación con los datos incorrectos</p>
Resultados obtenidos	<p>Cuando los datos correctos se envía un mensaje de datos correctos</p> <p>Cuando algunos de los datos no fueron correctos o vacíos, se mostró un mensaje informando el error y retornó al formulario para la corrección de ellos.</p>

	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017


2) Bases de Datos

El siguiente modelo de datos es un diseño de un sistema básico de reservaciones de habitaciones para un hotel



El diccionario de datos es el siguiente:

Table	field	Description
reservation	id	identificador de la reservación (Primary key)
	room_type_id	Identificador del tipo de habitación (Llave foránea)
	startdate	Fecha de inicio de la reservación
	enddate	Fecha de fin de la reservación
	admin_id	Identificador del usuario administrador que realiza la reserva
user	id	Identificador del usuario cliente
	firstname	Nombre del cliente
	lastname	Apellido del cliente
	phonenumber	Número de teléfono del cliente
	birthdate	Fecha de nacimiento del cliente
	email	correo electrónico del cliente
reservation_user	user_id	identificador del usuario cliente que realiza la reserva (Llave foránea)
	reservation_id	identificador de la reserva (Llave foránea)
rooms_type	id	identificador del tipo de habitación (Primary Key)
	name	nombre del tipo de habitación

	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017

	nof	Cantidad de habitaciones por tipo
--	-----	-----------------------------------

Los datos de ejemplo son los siguientes:

User

id	firstname	lastname	phonenummer	birthdate	email
1	Jhon	Doe	1-451-395-1600	1980-09-28	jdane@gmail.com
2	Jane	Jackson	1-709-896-0629	1985-05-01	jjackson@yahoo.com
3	Alex	Smith	1-742-823-9990	1990-05-28	jasmith@oul.com
4	Johana	Roll	1-806-460-9612	1981-10-31	jkrolling@uk.com

Room_type

id	name	nof
1	Single	2
2	Double	1
3	Shared	2

Reservation

id	room_type_id	startdate	enddate	admin_id
10	1	2017-04-29	2017-05-01	1
11	2	2017-04-29	2017-05-01	1
12	3	2017-04-29	2017-05-01	2
13	2	2017-05-02	2017-05-05	2
14	1	2017-05-06	2017-05-10	1
15	3	2017-05-26	2017-05-29	2
16	2	2017-05-26	2017-05-29	2
17	2	2017-05-06	2017-05-10	3

Reservation_user


user_id	reservation_id
1	10
2	11
3	12
4	13
1	14
2	15
3	16
4	17

Se requiere construir las siguientes consultas sql:

- Una consulta que permita determinar si para una fecha dada de inicio y fin por tipo de habitación existen habitaciones disponibles, la salida de la consulta debería ser como la siguiente:





Para la consulta realizada :

```
SET @numero1=0;
SET @numero2=0;
SET @numero3=0;
select room_type.id,startDate, endDate, nof, CASE room_type.id WHEN 1 THEN @numero1:=@numero1+1 WHEN 2 THEN @numero2:=@numero2+1
WHEN 3 THEN @numero3:=@numero3+1 ELSE 0 END AS num_of_reservation,CASE room_type.id WHEN 1 THEN nof-@numero1
WHEN 2 THEN nof-@numero2 WHEN 3 THEN nof-@numero3 ELSE 0 END AS available_rooms from reservation
JOIN room_type where startDate = '2017-04-29' and endDate = '2017-05-01' group by room_type.id;
```

	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017

13


<

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	id	startDate	endDate	nof	num_of_reservation	available_rooms
▶	1	2017-04-29	2017-05-01	2	1	1
	2	2017-04-29	2017-05-01	1	1	0
	3	2017-04-29	2017-05-01	2	1	1

room_type_id	startdate	enddate	num_of_rooms	num_of_reservations	available_rooms
1	2017-04-29	2017-05-01	2	1	1
2	2017-04-29	2017-05-01	1	1	0
3	2017-04-29	2017-05-01	2	1	1

Para el ejemplo anterior el administrador puede observar que no existen habitaciones tipo 2 (Double) entre el 29-04-2017 al 01-05-2017.

	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017

- b. Una consulta que permita determinar a un administrador a cuáles clientes ha atendido y la cantidad de reservas que el cliente ha realizado con el, la salida de la consulta debería ser como la siguiente:

```
SET @admin1=0;
SET @admin2=0;
SET @admin3=0;
SELECT A.admin_id, T.reservation_id, CASE admin_id WHEN 1 THEN @admin1:=@admin1+1 WHEN 2 THEN @admin2:=@admin2+1 WHEN 3 THEN @admin3:=@admin3+1 ELSE 0 END AS num_of_reservation , U.firstName, U.lastName FROM reservation A INNER JOIN Reservation_user T ON A.id = T.reservation_id INNER JOIN User u ON u.id = T.user_id order by A.admin_id ;
```

Result Grid Filter Rows: Export: Wrap Cell					
	admin_id	reservation_id	num_of_reservation	firstName	lastName
▶	1	10	1	Jhon	Doe
	1	14	2	Jhon	Doe
	1	11	3	Jane	Jackson
	2	15	1	Jane	Jackson
	2	12	2	Alex	Smith
	2	16	3	Alex	Smith
	2	13	4	Johana	Roll
	3	17	1	Johana	Roll


admin_id	num_of_reservations	firstname	lastname
1	1	Jane	Jackson
1	2	Jhon	Doe
2	2	Alex	Smith
2	1	Jane	Jackson
2	1	Johana	Roll
3	1	Johana	Roll

3) Desarrollo

Basado en un pseudo-framework que sigue el modelo MVC y utiliza POO escriba el código en PHP para una funcionalidad que permita a un administrador del hotel determinar si dada una fecha específica y un tipo de habitación existe disponibilidad para registrar la reserva. Para la implementación del método que accede a los datos utilice la consulta SQL (a) del punto anterior.

El funcionamiento del pseudo-framework es el siguiente: Existe un único punto de entrada que se llama index.php, desde el cual se invocan todos los controladores y acciones, cada controlador implementa sus acciones en métodos públicos, desde los controladores se acceden a las vistas y las vistas utilizan los modelos para obtener la información del repositorio de datos y envía la respuesta al controlador:

El siguiente es el código de un controlador de ejemplo:

	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017

```
<?php

use controllers;
use views;

class UsuariosController extends Controllers{

    public function loginAction () {
        //objeto genérico que procesa todas las salidas a la pantalla,
        //recibe el nombre de la vista
        $objView = new View("Usuarios");

        //método para pasar un modelo a la vista
        $objView->assignModel("Usuarios");

        //método para enviar a pantalla el resultado de una vista
        $objView->display();
    }
}
```

El siguiente es el código de una vista de ejemplo:

```
<?php

use views;
use models;
use request;

class UsuariosView extends Views{

    public function display () {
        //Obtiene el modelo enviado desde el controlador
        $objModel = $this->getModel("Usuarios");


        $objRequest = new Request("POST");
        $txtNombre = $objRequest->getParam("nombre")

        //utiliza el método para buscar un usuario por nombre,
        //recibe un array asociativo y retorna un array
        $arrUsuario = $objModel->findUsuario(array("nombre"=>$txtNombre));
        //Método para asignar un valor al template
        //Parámetro 1: nombre de la variable en el template
        //Parámetro 2: nombre del valor, array u objeto a asignar
        $this->assign("arrUsuario", $arrUsuario);

        //método para enviar a pantalla el resultado de una vista
        $this->out("login.tpl");
    }
}
```

En este ejemplo en el archivo "login.tpl" accede a la variable "arrUsuarios" como una variable php encerrado entre llaves {\$arrUsuarios["nombre"]}, en este punto solo se debe escribir etiquetas html.

```
<table>
    <tr>
        <td>Nombre:</td>
        <td>{$arrUsuario['name']}</td>
    </tr>
</table>
```


	Prueba Técnica de Ingreso - Ingeniero de Desarrollo		
	Versión 01	Fecha Creación 25/04/2017	Última modificación 25/04/2017

El siguiente es el código de ejemplo de un modelo:

```
<?php

use models;

class UsuariosModel extends Models{

    public function findUsuarios ($arrParametros){
        //permite asignar una consulta al gestor de base de datos
        $this->query("Select * from tbl_usuarios as u where u.nombre:= pnombre");
        //permite asignar el valor de parámetro "pnombre" a la consulta
        $this->param("pnombre", $txtNombre);
        //permite obtener el resultado de ejecutar la consulta
        return $this->getResult();
    }
}
```

De acuerdo a lo anterior usted debe crear:

- Un controlador para procesar la petición,
- Una vista que contenga el método que invoca al modelo y obtiene los datos
- Un template html para mostrar la información en pantalla
- Un modelo con el método para consultar la información,

Reutilice el código de ejemplo propuesto, tenga en cuenta la convención de nombres, dado que es el estándar del framework propuesto, utilice nombres de variables y clases adecuados.

El código fue enviado junto a la base de datos y la consultas realizadas anteriormente.