# **Formulario: Vista Lógica del Proyecto de Software**

**Proyecto:** Hosteluxe  
**Grupo:** 008D  
**Fecha:** 23-10-24

**1. Descripción General del Sistema**

Proporciona una descripción general del sistema y los principales componentes lógicos que lo forman.

**Descripción del sistema:**

La funcionalidad principal del software es agendar y reservar habitaciones en un hotel de forma rápida y segura, acompañada a esta, están las funcionalidades extras, que si bien no son principales si son igual de importantes que el componente principal(reserva). Y así formar una experiencia en base a la reserva, desde el momento en que se reserva hasta que el cliente se va del hotel queremos que sea algo único. Para eso se adhieren los siguientes componentes para complementar nuestra experiencia:

Relacionado al mismo componente de reserva se añade una forma de acercarnos al cliente en donde ellos nos dan información sobre ellos de forma implícita para poder darles una bienvenida mas cálida, esto no traspasa la barrera de su privacidad más allá de saber un poco de lo que le gusta, para eso se tienen preparadas diferentes opciones en base al gusto para dar una experiencia y lograr cautivar al cliente.

Para que el cliente se sienta como un rey, queremos que el haga el mínimo esfuerzo en su estadía para que logre un descanso pleno y un disfrute completo de su tiempo, por esto el cliente desde la misma pagina de reserva, una vez ingresado en el hotel el puede pedir servicio al cuarto donde habrá una diversa variedad en menús, desde comida nacional hasta italiana de la mas gran calidad que puede brindar la cafetería del hotel.

Detrás de todo esto se encuentra un sistema de control y gestión de todo el trafico del sistema, donde un supervisor puede ver todo lo que pasa en el día, donde los trabajadores pueden conocer sus tareas solo consultando el sistema, es cómodo y fácil de usar para cualquiera.

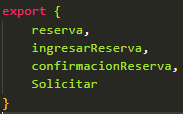
**2. Principales Clases y Entidades del Sistema**

Describe las clases o entidades principales del sistema, sus atributos y métodos principales, y su propósito dentro del sistema.

**Clase 1 (Reserva):   
Atributos:**

* **Id reserva**
* **Cantidad de días**
* **Fecha y hora de inicio y fin de la reserva**
* **Fecha de creación**
* **N° de habitación**
* **Usuario que reservo**
* **Monto total de la reserva**

**Métodos:**

**  
Propósito:** Con estos métodos se logra completar todo el sistema de reserva, con esto se hace ingreso a la BD de la información, para que así el cliente pueda agendar con nosotros

**Clase 2 (Experiencia):   
Atributos:**

* **Experiencia Id**
* **Tipo de experiencia**
* **Nombre de la experiencia/evento**
* **Asociación a la reserva**

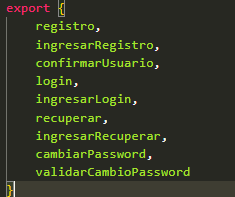
**Métodos: en este caso no tiene un controller como tal, pero se adhiere a los métodos de EmpleadoController y a ReservaController   
Propósito: Con esto logramos el objetivo de generar esta experiencia sin tener que estar generando código de mas**

**Clase 3 (Usuario):   
Atributos:**

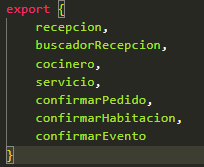
* **Id usuario**
* **Nombre y Apellidos**
* **Correo**
* **Contraseña**
* **Confirmado**
* **Foto de perfil**
* **Tipo de cuenta**
* **Teléfono**

**Métodos:**

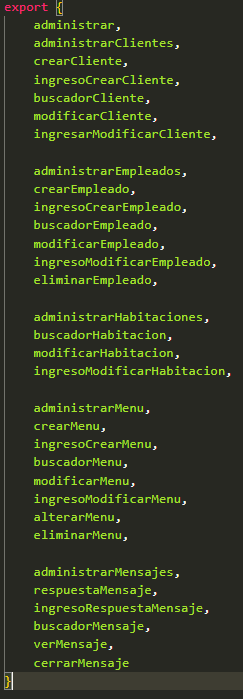
Controller usuario:

****

Controller Empleados:

****

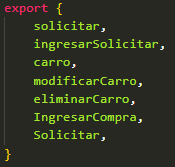
Controller de administracion:

**  
Propósito:** el propósito de cada uno de los métodos es manejar la información y dar diferentes funcionalidades, cada una fue nombrada de forma que se entienda que realiza dentro del software.

**Clase 4 (Pedidos Menú/Cafetería):   
Atributos:**

* **Id del pedido**
* **Id del menú**
* **Nombre del plato**
* **Descripción**
* **Precio**
* **Estado**
* **Categoría**

**Métodos:**

**  
Propósito: principalmente es el de ingresar un pedido, mandarlo a cocina, que se prepare para luego mandarlo con el servicio al cuarto y realizar la entrega del pedido en el cuarto donde se hospeda el cliente.**

**3. Relaciones entre clases**

Especifica cómo están relacionadas las clases entre sí, indicando relaciones de herencia, composición, agregación o dependencias.

**Relaciones entre clases:**

* **Reserva: Se relaciona con las tablas de habitación, usuario y evento. Heredando sus Id**
* **Habitación: se relaciona con el modelo Tipos, heredando los tipos de habitación que contiene la tabla**
* **Usuario: se relaciona con cuentas, similar a tipos, hereda los tipos de cuenta (recepción, cliente, administrador, etc.)**
* **Evento: se relaciona con tipo evento**
* **Pedido:**
* **Sub Categoría: con el modelo de menú y con el modelo de categoría, heredando los tipos de platos, platos, etc.**
* **Todo el detalle de las herencias y relaciones:**



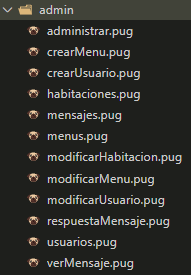
**4. Interfaces y dependencias**

Indica si alguna de las clases o entidades tiene interfaces y cómo interactúan con otros módulos del sistema a través de ellas.

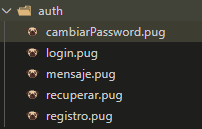
**Descripción de interfaces y dependencias:**

**Interfaces por clases:**

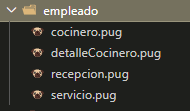
* **Vistas de administración:**

****

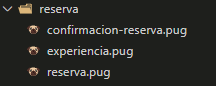
* **Vistas de login (auth):**

****

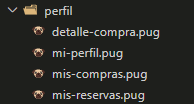
* **Vistas de los trabajadores/ empleados:**

****

* **Vistas asociadas a la reserva:**

****

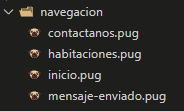
* **Vistas asociadas al perfil:**

****

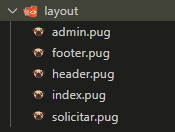
* **Vistas de los pedidos/ menú / cafetería:**

****

* **Vistas varias de navegación de usuario:**

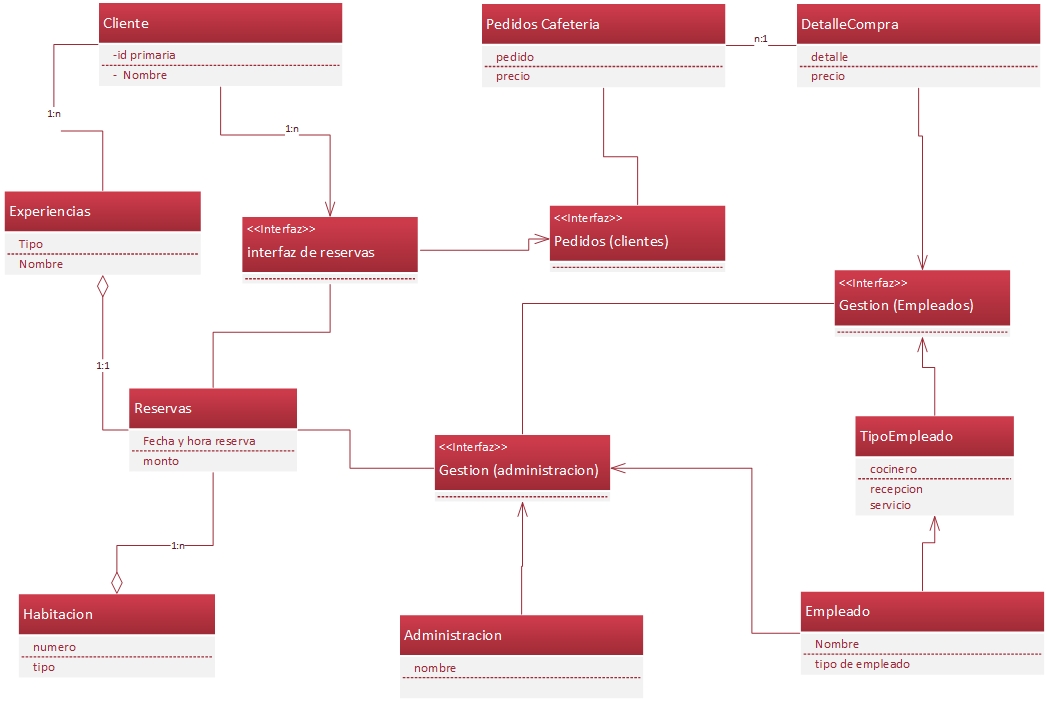
****

* **Layout de footer, header y diferentes módulos:**

****

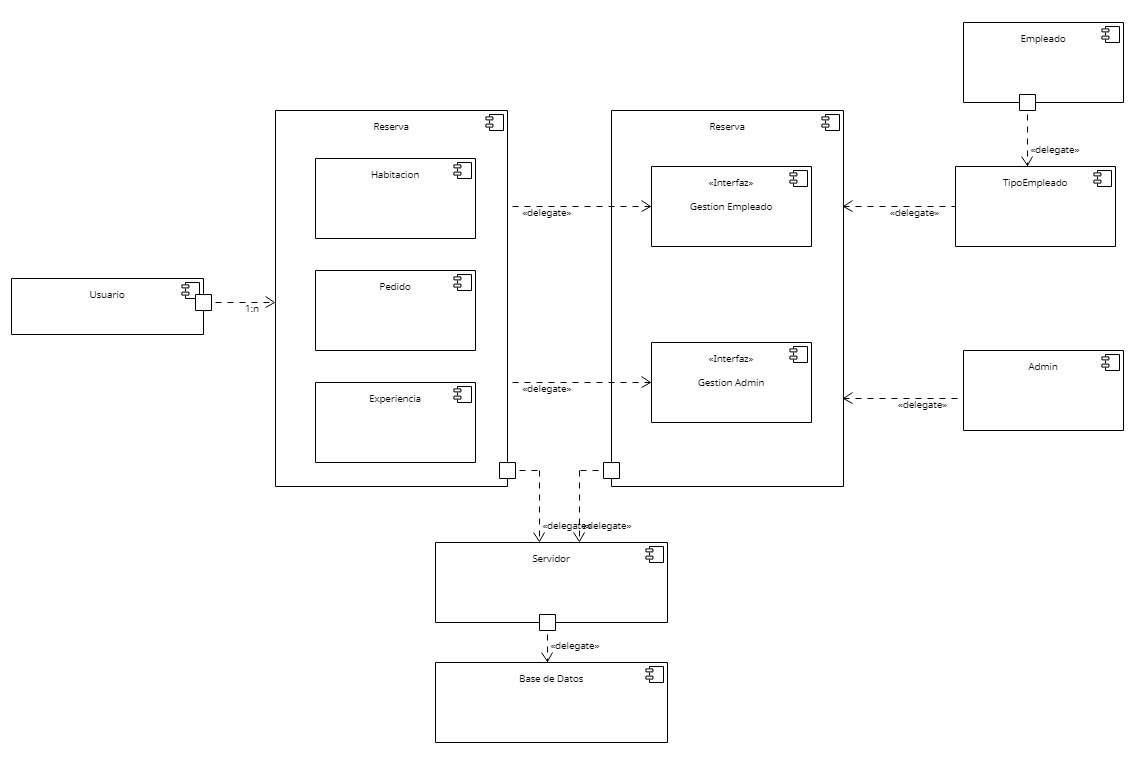
**5. Diagrama de Clases del Sistema**

Adjunta un diagrama de clases que representa la estructura lógica del sistema, las clases principales y las relaciones entre ellas.

**Diagrama de clases:****

**6. Diagrama de componentes (opcional)**

Si su sistema lo requiere, adjunte un diagrama de componentes que represente cómo los componentes principales del sistema están organizados y cómo interactúan.

**Diagrama de componentes:****

**7. Comentarios adicionales**

Incluye cualquier comentario adicional o aclaración relevante sobre la vista lógica de tu sistema.

**Comentarios:**