

Definición del Proyecto: Aplicación de Reservas Luxor

**Alumnos:** Nicolas Noimann

Belen Verdugo

Vicente Lopez

**Profesor:** Roberto Osses

**Asignatura:** Programacion de aplicaciones moviles

10 de Septiembre de 2024

Descripción general del proyecto

La aplicación de reservas Luxor está diseñada para mejorar la experiencia del cliente en la cadena hotelera de lujo Luxor. La aplicación proporcionará una plataforma centralizada para la gestión de reservas, acceso a servicios adicionales y un programa de lealtad. La implementación se llevará a cabo utilizando Ionic, permitiendo una experiencia nativa y responsive en diversas plataformas móviles.

Estructura del proyecto

**2. Estructura del Proyecto**

**2.1 Capa de Presentación**

**Objetivo:** Proporcionar una interfaz intuitiva y atractiva para los usuarios que permita una experiencia fluida en el proceso de reserva y gestión de servicios.

**Características:**

* **Interfaz de Usuario (UI):** Implementada con Ionic para garantizar una experiencia consistente en diferentes dispositivos y plataformas. Uso de componentes de UI como formularios, botones, menús y pantallas modales.
* **Experiencia de Usuario (UX):** Diseño centrado en el usuario, con navegación fácil entre vistas y una experiencia de reserva simplificada. Integración de elementos visuales que reflejen el lujo y la exclusividad de la cadena hotelera.
* **Responsive Design:** Adaptación a diferentes tamaños de pantalla y dispositivos (móviles, tablets).
* **Feedback en Tiempo Real:** Mensajes de confirmación y notificaciones para interacciones del usuario, como éxito en la reserva, errores de validación o actualizaciones en el estado de la reserva.

**2.2 Capa de Lógica de Negocio**

**Objetivo:** Implementar las reglas de negocio que gestionan el flujo de reservas, la disponibilidad de habitaciones, y la interacción con servicios adicionales y el programa de lealtad.

**Reglas de Negocio:**

* **Validación de Disponibilidad:** Verificación en tiempo real de la disponibilidad de habitaciones para las fechas seleccionadas.
* **Políticas de Reserva:** Aplicación de políticas específicas como reservas mínimas y máximas, cancelaciones y cambios de reserva, y condiciones de pago.
* **Servicios Adicionales:** Gestión de la disponibilidad y reserva de servicios adicionales (spa, restaurantes) y su integración con la reserva principal.
* **Promociones y Ofertas:** Aplicación de descuentos o promociones especiales basadas en fechas, temporadas, o estatus del cliente (nuevos clientes, clientes antiguos, etc.).
* **Autenticación y Autorización:** Gestión de acceso a funciones de la aplicación basadas en el rol del usuario (cliente, administrador) y la verificación de credenciales.

**Flujo de Datos:**

* **Reserva:** Los datos de la reserva se envían desde la capa de presentación a la lógica de negocio, que verifica la disponibilidad y aplica las políticas antes de confirmar la reserva y generar el recibo.
* **Actualización de Perfil:** Los cambios en el perfil de usuario se gestionan y se reflejan en la base de datos a través de la capa de acceso a datos.
* **Acceso a Servicios:** Las solicitudes para servicios adicionales se gestionan, validan y procesan en esta capa antes de ser confirmadas y añadidas a la reserva.

**2.3 Capa de Acceso a Datos**

**Objetivo:** Gestionar la interacción con la base de datos MySQL LIVE y las APIs para proporcionar la información necesaria a la capa de lógica de negocio.

**Componentes:**

* **Base de Datos (MySQL):**
  + **Tablas:** Estructura para almacenar información sobre usuarios, reservas, habitaciones, servicios adicionales, y puntos de fidelidad.
  + **Consultas:** Operaciones para insertar, actualizar, eliminar y consultar datos relacionados con reservas, disponibilidad de habitaciones, y gestión de servicios.
  + **Transacciones:** Manejo de transacciones para asegurar la integridad de los datos en operaciones críticas, como la reserva de habitaciones y la actualización de perfiles.
* **APIs:**
  + **API de Reserva:** Endpoints para crear, consultar, modificar y cancelar reservas.
  + **API de Disponibilidad:** Endpoints para verificar la disponibilidad de habitaciones y servicios adicionales.
  + **API de Usuario:** Endpoints para gestionar la autenticación, registro, y actualización de perfiles de usuario.
  + **API de Lealtad:** Endpoints para gestionar el programa de lealtad, incluyendo la acumulación de puntos y la consulta de recompensas.

**\*Nota:** Las API´s que se utilizaran en este proyecto quedaran en espera mientras se ven las implementaciones técnicas del proyecto. \*

**Integración:**

* **Consultas en Tiempo Real:** Interrogación a la base de datos para obtener la disponibilidad actualizada y gestionar reservas en tiempo real.
* **Sincronización de Datos:** Actualización y sincronización de datos entre la base de datos y la capa de presentación para asegurar que la información mostrada al usuario sea precisa.

**2.4 Consideraciones Adicionales**

* **Seguridad:** Implementación de medidas de seguridad para proteger la información personal y de pago de los usuarios. Uso de HTTPS para la comunicación segura y encriptación de datos sensibles.
* **Escalabilidad:** Diseño del sistema para permitir la expansión a nuevas ubicaciones y servicios sin requerir una reestructuración significativa.
* **Mantenimiento:** Utilización de una arquitectura modular para facilitar el mantenimiento y la actualización del sistema a medida que cambian los requisitos del negocio o se agregan nuevas características.

Este diseño integral asegura que la aplicación de reservas Luxor no solo cumpla con las expectativas de los clientes en términos de funcionalidad y lujo, sino que también ofrezca una solución escalable y mantenible para el negocio.

**3. Vistas de Pantalla**

**3.1 Inicio:**

* + **Objetivo:** Ofrecer una visión general del sistema y acceso a las principales funcionalidades.
  + **Funcionalidades:**
    - Presentación de ofertas y promociones destacadas.
    - Acceso a servicios adicionales (spa, restaurantes).
    - Información general sobre la cadena hotelera.
    - Botones de navegación hacia las otras vistas (Ingreso de datos, Compra, Contacto, Perfil de usuario, Login).
  1. **Ingreso de Datos:**
  + **Objetivo:** Permitir a los usuarios ingresar la información necesaria para realizar una reserva.
  + **Funcionalidades:**
    - Formulario para ingresar detalles de la reserva (fechas, tipo de habitación, número de huéspedes).
    - Opciones para agregar servicios adicionales (spa, restaurante).
    - Validación de datos y disponibilidad en tiempo real.

**3.3 Compra:**

* + **Objetivo:** Facilitar el proceso de compra y confirmación de reservas.
  + **Funcionalidades:**
    - Resumen de la reserva y servicios seleccionados.
    - Opción para ingresar datos de pago.
    - Confirmación de reserva y generación de recibo.
    - Integración con un sistema de pago seguro.

**3.4 Contacto:**

* + **Objetivo:** Proveer canales de comunicación entre los clientes y el equipo de soporte.
  + **Funcionalidades:**
    - Formulario de contacto para consultas o soporte.
    - Información de contacto de la cadena hotelera (teléfonos, correos electrónicos, redes sociales).
    - Opciones para enviar mensajes o agendar llamadas.
  1. **Perfil de Usuario:**
  + **Objetivo:** Permitir a los usuarios gestionar su información personal y preferencias.
  + **Funcionalidades:**
    - Visualización y edición de información personal (nombre, dirección, preferencias).
    - Consulta del historial de reservas y puntos de fidelidad.
    - Opciones para actualizar la contraseña y configurar notificaciones.

**3.6 Login:**

* + **Objetivo:** Autenticar a los usuarios y permitir el acceso a funcionalidades personalizadas.
  + **Funcionalidades:**
    - Formulario de inicio de sesión (correo electrónico y contraseña).
    - Opciones para recuperar contraseña o crear una nueva cuenta.
    - Integración con el sistema de autenticación para verificar credenciales.

**3.7 Tecnologías Utilizadas**

* **Frontend:** Ionic Framework para el desarrollo de una interfaz de usuario responsiva y nativa.
* **Backend:** API REST para la gestión de reservas y acceso a datos.
* **Base de Datos MySQL Live:** Sistema de gestión de base de datos relacional para almacenar información de reservas y usuarios.
* **Seguridad:** Protocolos de seguridad nativas de IONIC para la protección de datos y transacciones.

**4. Consideraciones Adicionales**

* **Escalabilidad:** El sistema está diseñado para permitir la incorporación de nuevas ubicaciones y servicios sin comprometer la estabilidad general.
* **Mantenimiento:** La estructura en capas facilita el mantenimiento y la actualización de la aplicación.
* **Experiencia de Usuario:** Se prioriza una interfaz intuitiva y accesible, con una navegación fluida entre vistas.

Este enfoque garantiza que la aplicación de reservas Luxor cumpla con los estándares de una aplicación web del mercado y ofrezca una experiencia de usuario decente, mientras se adapta a las necesidades futuras del negocio.