

Flujo de navegación de los clientes.

Nota: Mantener API de validarDisponibilidadXXX(), aunque no se usan ahora; definir de forma similar a las funciones de búsqueda y reserva (ID del producto/servicio, fecha de inicio, fecha de fin, número de personas).

1. El cliente ingresa en la página.

Respuesta: El sistema muestra la página principal.

2. El cliente inicia sesión.

Respuesta: El sistema valida los datos con la base de datos y establece una cookie de sesión iniciada.

Base de datos: Usuarios: Correo electrónico, Nombre, Apellido, Dirección física, (País), Edad, Tipo de identificación (DNI, pasaporte, SSN), Identificación.

3. El cliente explora el catálogo.

Respuesta: El sistema carga todos los servicios del catálogo y muestra sus títulos, precios e imágenes.

Base de datos: Servicios: URI, Tipo de servicio, Nombre.

API: Obtener servicios (como buscarXXX(), que debería incluir las URI de las imágenes dentro de la respuesta). Los servicios incluyen un precio estándar y un precio actual que, si es diferente, indica un descuento o un incremento de precio. **Nota:** Al referirnos a "Servicios" hablamos de los productos que ofrecen los proveedores, como vuelos o habitaciones de hotel.

Nota: Datetime y Timestamp usa la serialización por defecto de ASP.NET. Estos se guardan como un String en JSON.

API Aerolínea: Todos los parámetros opcionales, para que devuelva todos los vuelos disponibles. Parámetros opcionales: origen (from) (String), destino (to) (String), fecha de despegue (date_from) (Datetime), fecha de llegada (date_to) (Datetime), tipo de cabina (cabin) (String: *economica, premium*), número asientos disponibles para pasajeros en la cabina (pasajeros) (Int32), opción de ordenamiento (sort) (String: *POR DEFINIR*), precio mínimo (precio_min) (Decimal), precio máximo (precio_max) (Decimal).

Retorno: Lista de Vuelo: {id_vuelo (String), origen (String), destino (String), fecha (Datetime), tipo_cabina (String), capacidad_pasajeros (Int32), nombre_areolinea (String), capacidad_actual (en el tipo de cabina) (Int32), precio_normal (Decimal), precio_actual (Decimal)}

API Hoteles: Todos los parámetros opcionales, para que devuelva todas las habitaciones disponibles. Parámetros opcionales: origen (from) (String), destino (to) (String), fecha de llegada (date_from) (Datetime), fecha de salida (date_to) (Datetime), tipo de habitación (tipo_habitacion) (String: *POR ACLARAR*), número plazas disponibles para inquilinos (capacidad) (Int32), precio mínimo (precio_min) (Decimal), precio máximo (precio_max) (Decimal).

Retorno: Lista de Habitación: {id_habitacion (String), ciudad (String), fechas (Datetime), tipo (String), capacidad (Int32), precio_normal (Decimal), precio_actual (Decimal), uri_imagen (String)}; dato adicional (fuera de la lista): ciudad (String), pais (String).

API Autos: Todos los parámetros opcionales, para que devuelva todos los vehículos disponibles. Parámetros opcionales: ciudad (String), pais (String), fecha de inicio de alquiler (desde) (Datetime), fecha de fin de alquiler (hasta) (Datetime), categoría (categoria) (String: *POR DEFINIR*), tipo de transmisión (transmision) (String: *POR DEFINIR*),

capacidad de pasajeros (capacidad) (Int32), precio mínimo (precio_min) (Decimal), precio máximo (precio_max) (Decimal), opción de ordenamiento (sort) (String: POR DEFINIR).

Retorno: Lista de Autos: {**id_auto (String)**, **ciudad (String)**, **fechas (Datetime)**, tipo (String), **capacidad (Int32)**, precio_normal (Decimal), precio_actual (Decimal), **uri_imagen (String)**}; dato adicional (fuera de la lista, si solo es en un lugar; de lo contrario, adentro de la lista, con cada vehículo): **ciudad (String)**, **pais (String)**.

API Paquetes: Todos los parámetros opcionales, para que devuelva todos los paquetes disponibles. Parámetros opcionales: **ciudad (String)**, **pais (pais) (String)**, fecha de inicio de paquete (**fecha_inicio**) (Datetime), duración de la experiencia (duracion) (Timestamp), tipo de actividad (tipo_actividad) (String: POR DEFINIR), **capacidad de turistas (capacidad) (Int32)**, precio mínimo (precio_min) (Decimal), precio máximo (precio_max) (Decimal), opción de ordenamiento (sort) (String: POR DEFINIR).

Retorno: Lista de Paquete: {**id_paquete (String)**, **ciudad (String)**, **pais (String)**, tipo (String), **capacidad (Int32)**, precio_normal (Decimal), precio_actual (Decimal), uri_imagen (o URL de información adicional del paquete, por definir) (String)}.

API Restaurantes: Todos los parámetros opcionales, para que devuelva todas las mesas disponibles. Parámetros opcionales: **ciudad (String)**, **pais (pais) (String)**, **fecha reserva (fecha) (Datetime)**, categoria (categoria) (String: POR DEFINIR), **tipo de mesa (tipo) (String: POR DEFINIR)**, **puestos a la mesa (capacidad/disponibilidad) (Int32)**, precio mínimo (precio_min) (Decimal), precio máximo (precio_max) (Decimal), opción de ordenamiento (sort) (String: POR DEFINIR).

Retorno: Lista de Mesas: {**id_mesa (String)**, **ciudad (String)**, tipo (String), categoria (String), **capacidad (Int32)**, precio_normal (Decimal), precio_actual (Decimal), **uri_imagen (String)**}; dato adicional fuera de la lista: **ciudad (String)**, **pais (String)**.

4. El cliente filtra según sus intereses.

Respuesta: El sistema recarga solo los servicios que correspondan con el tipo de servicio seleccionado y con otros filtros. Vuelve a invocar las API hasta que se establezca una caché.

Base de datos: Servicios: URI, Tipo de servicio, Nombre.

API: Obtener imágenes, Obtener servicios; hasta que se establezca caché (o se puede guardar en memoria al cargar la página del catálogo, hasta que se recargue).

5. El cliente expande detalles del servicio.

Respuesta: El sistema aporta más información del servicio, como precio o fecha de salida.

Base de datos: Servicios: URI, Tipo de servicio, Nombre, con SELECT del URI o ID específico.

API: Obtener servicios (guardado de la última búsqueda, pero estaría bien una API que permitiera la búsqueda individual).

6. El cliente añade el servicio al carrito

Respuesta: El sistema añade el intento de reserva al carrito, que se guarda o bien en el almacenamiento del navegador o en la base de datos, en sección de prerreservas.

Base de datos: Carrito o Prerreservas (si se guarda en la BDD): Servicio, URI de imagen, Detalles, como precio y fechas (o bien una cadena de texto JSON, o bien una cantidad abstracta de datos, por determinar).

7. El cliente continúa explorando.

Respuesta: Ver anteriores.

8. El cliente abre su carrito.

Respuesta: El sistema carga todos los elementos del carrito.

Base de datos: Ver punto anterior.

9. El cliente hace clic en comprar.

Respuesta: El sistema invoca el método de creación de prerreservas de cada servicio y presenta la pestaña de pagos, que se conecta con la API del banco; escribirá un número de cuenta de donde se debitarán los valores.

Base de datos: Servicios, Carrito o Prerreservas.

API: Bloqueo de prerreservas (`crearPreReservaXXX()`); este método debería invocar también al que valida la disponibilidad (`validarDisponibilidadXXX()`).

API Aerolínea: `crearPreReservaVuelo()` recibe la `id_vuelo` y un arreglo de pasajeros (`pasajeros {nombre, apellido, fecha_nacimiento, tipo_identificacion, identificacion}`); opcionalmente, se agrega una duración de Hold (`duracionHoldSegundos`) con un valor por defecto de 600 segundos. Devuelve la ID del Hold (`id_hold`).

API Hoteles: `crearPreReservaHabitacion()` recibe la `id_habitacion (String)`, el **número de huéspedes (numero_huespedes) (Int32)**, la **fecha de llegada (fecha_llegada) (Datetime)** y la **fecha de salida (fecha_salida)**; opcionalmente, se agrega una duración de Hold (`duracionHoldSegundos`) con un valor por defecto de 600 segundos. Devuelve la ID del Hold (`id_hold`).

API Autos: `crearPreReservaAuto()` recibe la `id_vehiculo (String)`, la **fecha de inicio del alquiler (fecha_inicio) (Datetime)** y la **fecha de fin del alquiler (fecha_fin)**; opcionalmente, se agrega una duración de Hold (`duracionHoldSegundos`) con un valor por defecto de 600 segundos. Devuelve la ID del Hold (`id_hold`).

API Paquetes: `crearPreReservaPaquete()` recibe la `id_paquete (String)` y un arreglo de turistas (`turista {nombre, apellido, fecha_nacimiento, tipo_identificacion, identificacion}`); opcionalmente, se agrega una duración de Hold (`duracionHoldSegundos`) con un valor por defecto de 600 segundos. Devuelve la ID del Hold (`id_hold`).

API Restaurantes: `crearPreReservaAuto()` recibe la `id_mesa (String)`, la **fecha de reserva (fecha) (Datetime)**, y el **número de personas (numero_clientes) (Int32)**; opcionalmente, se agrega una duración de Hold (`duracionHoldSegundos`) con un valor por defecto de 600 segundos. Devuelve la ID del Hold (`id_hold`).

10. El cliente realiza el pago.

Respuesta: El sistema invoca los métodos de transacción del banco, y registra las reservas en cada API involucrada y en la base de datos. Luego, permite imprimir una factura (los valores de cada reserva correspondiente también son transferidos a los dueños de los servicios).

Base de datos: Usuarios, Servicios, Carrito o Prerreservas, Reservas: Fecha, Valor debitado, Detalles de la reserva (véase Prerreserva). La reserva es individual para cada servicio; adicionalmente, se puede crear una tabla de Pagos que reúna todas las reservas realizadas con una transacción.

API: Acción de reservar (`reservarXXX()`), Acción de creación de clientes (no usuarios, aunque esté definida como `crearUsuarioExterno()`), y devuelve una identificación ya

existente si ya estaba registrado ese cliente, o una nueva si no existía; en cualquier caso, los proveedores de servicios pueden decidir si es relevante guardar esa información como un registro único o solo usarlo para facturación y administración de reservas). Se conecta con la API del banco para realizar el débito de la cuenta del cliente (crédito a la cuenta del bus de integración), y luego se debita de la cuenta del bus de integración para acreditar a todos los proveedores de servicios, menos una comisión. No se invoca una API de generación de factura, porque la factura del cliente se realiza con el bus de integración.

Paso 1: Generar reserva:

API Aerolínea: id_vuelo (String), id_hold (String), arreglo de pasajeros (pasajeros) (definido anteriormente), correo de la persona a la que va dirigida la reserva (correo) (String).

API Hoteles: id_habitacion (String), id_hold (String), número de huéspedes (numero_huespedes) (Int32), nombre del titular de la reserva (nombre) (String), apellido (String), tipo de identificación (tipo_identificacion) (String), identificación (identificacion) (String), correo de la persona a la que está a nombre la reserva (correo) (String), fecha de llegada (fecha_llegada) (Datetime), fecha de salida (fecha_salida) (Datetime).

API Autos: id_auto (String), id_hold (String), nombre del titular de la reserva (nombre) (String), apellido (String), tipo de identificación (tipo_identificacion) (String), identificación (identificacion) (String), correo de la persona a la que está a nombre la reserva (correo) (String), fecha de inicio del préstamo (fecha_inicio) (Datetime), fecha de fin del préstamo (fecha_fin) (Datetime).

API Paquetes: id_paquete (String), id_hold (String), correo de la persona a la que está a nombre la reserva (correo) (String), arreglo de turistas (turistas) (definido previamente).

API Restaurantes: id_mesa (String), id_hold (String), nombre del titular de la reserva (nombre) (String), apellido (String), correo de la persona a la que está a nombre la reserva (correo) (String), fecha de la reservación (fecha) (Datetime), número de personas (incluido el que reserva) (personas) (Int32).

Paso 2: Generar factura: Todos devuelven la URI de la factura generada, para poder acceder a ella en cualquier momento (formato PDF, HTML, etc.):

Todas reciben el ID de la reserva (id_vuelo, id_habitacion, etc.) (String), el correo de la persona a nombre de la que se hace la factura (correo) (String), el nombre de la persona para la que se hace la factura (nombre) (String), el tipo de identificación (tipo_identificacion) (String), la identificación de la persona (identificacion) (String), el valor pagado (valor) (Decimal), la ID de la transacción de la API del banco (id_transaccion) (String).

11. El cliente ve las reservas en su perfil.

Respuesta: El sistema carga las reservas guardadas en la base de datos.

Base de datos: Reservas, Servicios, Usuarios.

API: Todas las API: buscarDatosReserva(), que recibe id_reserva.

API Aerolínea: Devuelve origen, destino, correo, fecha, tipo_cabina, arreglo de pasajeros (definido previamente), nombre_aerolinea, asientos_reservados, valor_pagado, uri_factura.

API Hoteles: Devuelve numero_habitacion, correo, fecha_llegada, fecha_salida, tipo_habitacion, numero_huespedes, **valor_pagado**, **uri_factura**.

API Autos: Devuelve numero_matricula, correo, fecha_inicio, fecha_fin, categoria, transmision, **valor_pagado**, **uri_factura**.

API Paquetes: Devuelve id_paquete, correo, fecha_inicio, duracion, tipo_actividad, arreglo de turistas (turistas) (véase lo definido previamente), **valor_pagado**, **uri_factura**.

API Restaurantes: Devuelve numero_mesa, correo, fecha, numero_personas, categoria, tipo, nombre de la persona que reservó (nombre), apellido, **valor_pagado**, **uri_factura**.

Nota: El sistema no permite la cancelación de reservas.