**Clases Principales**

1. **HabitacionBean**
   * **Propiedades**: id, identificador, plazas, planta, observaciones, estado, camas, precioAdultos, precioMenores.
   * **Métodos**: Getters y setters para todas las propiedades, métodos para convertir a JSON, y constructores.
2. **HabitacionParcialBean**
   * **Propiedades**: idHabitacion, idHabitacionRaiz, estado.
   * **Métodos**: Getters y setters, y constructores.
3. **InscritoBean**
   * **Propiedades**: nombre, apellido, telefono, email, telegram, nif, id, pseudonimo, estado, menor, alimentos, alimentos\_txt, alergias, alergias\_txt, observaciones, habitacion, smial, con\_bebes.
   * **Métodos**: Getters y setters, métodos para calcular importe y obtener tipo de habitación, y constructores.
4. **MensajeBean**
   * **Propiedades**: from, to, copyto, blindcopyto, subject, body, replyto.
   * **Métodos**: Getters y setters, y un método para generar un mensaje MIME.
5. **NoticiaBean**
   * **Propiedades**: idNoticia, titulo, cuerpo, estado, fecha, fechaUpdate.
   * **Métodos**: Getters y setters, métodos para formatear fecha y convertir a JSON, y constructores.
6. **UsuarioBean**
   * **Propiedades**: id, usuario, password, idInscrito.
   * **Métodos**: Getters y setters.

**Funcionalidades Adicionales**

* **LoggerPublic**: Clase para gestionar logs de errores y actividades.
* **SQLHabitacionesPublic**: Métodos para interactuar con la base de datos relacionada con habitaciones.
* **SQLInscritosPublic**: Métodos para gestionar inscritos en la base de datos.
* **SQLNoticiasPublic**: Métodos para gestionar noticias en la base de datos.
* **SQLPlanificacionPublic**: Métodos para gestionar la planificación de actividades.
* **SQLUsuariosPublic**: Métodos para gestionar usuarios en la base de datos.

**Servlets**

* **ListNoticiasPublish**: Servlet para listar noticias.
* **ViewNoticia**: Servlet para ver una noticia específica.
* **ActividadesSala**: Servlet para gestionar actividades en una sala.
* **BloquearHabitacion**: Servlet para bloquear una habitación.
* **BloquearHabitacionParcial**: Servlet para bloquear parcialmente una habitación.
* **CheckInscrito**: Servlet para verificar si un inscrito existe.
* **CheckStatusPlazas**: Servlet para verificar el estado de las plazas.
* **GetComboHabitacionesDisponibles**: Servlet para obtener un combo de habitaciones disponibles.
* **RegistrarInscripcion**: Servlet para registrar una inscripción.
* **CheckStatusFrontend**: Servlet para verificar el estado del frontend.

**1. Formulario de Inscripción**

**Título**: Bienvenido a la hoja de inscripción de la Mereth Aderthad

**Descripción**: Página principal para la inscripción al evento Mereth Aderthad.

**Elementos**:

* **Encabezado**: Incluye el título de la página y enlaces a hojas de estilo y scripts.
* **Consentimiento de Cookies**: Script para gestionar el consentimiento de cookies utilizando TermsFeed.
* **Google Analytics**: Script para integrar Google Analytics.
* **Menú Principal**: Contiene botones para diferentes tipos de inscripción (General, Aleatoria, Lista de Espera) y un botón para reiniciar el proceso.
* **Pasos de Inscripción**: Divs (paso1, paso2, paso3, paso3individual, paso3espera, paso4) para guiar al usuario a través del proceso de inscripción.
* **Scripts**: Incluye jQuery y un script personalizado (mereth.js) para manejar la lógica de la inscripción.

**2. Términos y Condiciones**

**Título**: Términos y Condiciones

**Descripción**: Página que detalla los términos y condiciones para la inscripción al evento.

**Contenido**:

* **Derechos de Imagen**: Información sobre la toma y uso de imágenes durante el evento.
* **Condiciones de Reserva**: Detalles sobre el pago, confirmación, cancelaciones, y cambios de reserva.
* **Normas de Comportamiento**: Reglas que los participantes deben seguir durante el evento.
* **Exención de Responsabilidad**: Declaración de exención de responsabilidad por parte de la organización del evento.

**3. Control de Accesos de Asamblea General**

**Título**: Bienvenido al Control de Accesos de Asamblea General

**Descripción**: Página para el control de accesos mediante escaneo de códigos QR.

**Elementos**:

* **Encabezado**: Incluye el título de la página y enlaces a hojas de estilo y scripts.
* **Escaneo de QR**: Sección para escanear códigos QR y verificar la información de los socios.
* **Carga de Archivo CSV**: Permite cargar un archivo CSV con la información de los socios.
* **Scripts**: Incluye jQuery y un script para manejar el escaneo de QR (qr-scanner.min.js) y procesamiento de datos.

### Análisis de Interacción entre HTML y Java

#### HTML Forms y Servlets

1. **Formulario de Inscripción**:
   * **HTML**: Contiene varios formularios y botones para la inscripción general, inscripción aleatoria, y lista de espera.
   * **Servlets**:
     + RegistrarInscripcion: Maneja la inscripción de usuarios.
     + CheckInscrito: Verifica si un inscrito ya existe.
     + BloquearHabitacion: Bloquea una habitación durante el proceso de inscripción.
     + BloquearHabitacionParcial: Bloquea parcialmente una habitación.
2. **Términos y Condiciones**:
   * **HTML**: Página estática que muestra los términos y condiciones del evento.
   * **Servlets**: No interactúa directamente con servlets, pero proporciona información relevante para los usuarios.
3. **Control de Accesos de Asamblea General**:
   * **HTML**: Página para el control de accesos mediante escaneo de códigos QR.
   * **Servlets**:
     + CheckStatusPlazas: Verifica el estado de las plazas disponibles.
     + GetComboHabitacionesDisponibles: Obtiene una lista de habitaciones disponibles.

### Diagramas de Secuencia

#### Diagrama de Secuencia para el Proceso de Inscripción

@startuml

actor User

participant "HTML Form" as Form

participant "RegistrarInscripcion Servlet" as RegistrarInscripcion

participant "SQLInscritosPublic.altaInscrito" as AltaInscrito

participant "SQLHabitacionesPublic.estadoHabitaciones" as EstadoHabitaciones

User -> Form: Submit Inscription Form

Form -> RegistrarInscripcion: Call RegistrarInscripcion Servlet

RegistrarInscripcion -> AltaInscrito: Call altaInscrito Method

AltaInscrito --> RegistrarInscripcion: Return Result

RegistrarInscripcion -> EstadoHabitaciones: Call estadoHabitaciones Method

EstadoHabitaciones --> RegistrarInscripcion: Return Result

RegistrarInscripcion --> Form: Return Response

Form --> User: Display Confirmation

@enduml

#### Diagrama de Secuencia para el Control de Accesos

@startuml

actor User

participant "HTML Form" as Form

participant "CheckStatusPlazas Servlet" as CheckStatusPlazas

participant "SQLHabitacionesPublic.checkPlazasTotales" as CheckPlazasTotales

User -> Form: Load Access Control Page

Form -> CheckStatusPlazas: Call CheckStatusPlazas Servlet

CheckStatusPlazas -> CheckPlazasTotales: Call checkPlazasTotales Method

CheckPlazasTotales --> CheckStatusPlazas: Return Result

CheckStatusPlazas --> Form: Return Status

Form --> User: Display Status

@enduml

