**MANUAL TÉCNICO DE UN PROTOTIPO DE SOFTWARE PARA EL REGISTRO DE**

**RESERVACIONES EN UN HOTEL (UNIWAOS HOTEL)**

**DANIEL ESTEBAN RIVERA GÓMEZ**

**NATALIA LORENA RÍOIS CALDERÓN**

**MICHAEL STEVEN NARANJO BAUTISTA**

**UNIVERSIDAD DE CUNDINAMARCA**

**ING. SISTEMAS**

**FACATATIVÁ**

**2024**

**Tabla de contenido**

**Introduccion…………………………………………………………………………………….3**

**Objetivo del programa…………………………………………………………………………4**

**Componentes……………………………………………………………………………………5**

**Interaccion entre componenetes……………………………………………………………….6**

**Clases…………………………………………………………………………………………… 7**

**Capacidades del sistema………………………………………………………………………11**

**Introduccion**

El presente manual técnico describe el funcionamiento y la estructura del sistema de reservaciones desarrollado en Java utilizando el entorno de desarrollo NetBeans. Este sistema permite gestionar reservaciones de habitaciones a través de una estructura de árboles, listas y ArrayLists, proporcionando una solución eficiente y escalable para la administración de habitaciones y reservas.

El objetivo principal de este sistema es facilitar la gestión de reservas y habitaciones en hoteles u otros alojamientos, permitiendo a los administradores realizar un seguimiento detallado y ordenado de las reservas y la disponibilidad de las habitaciones a lo largo del tiempo. El sistema se ha diseñado con un enfoque modular para permitir futuras expansiones y mejoras.

**Objetivo del programa**

El objetivo principal del sistema de reservaciones es proporcionar una herramienta eficiente y fácil de usar para la gestión de reservas y disponibilidad de habitaciones en establecimientos hoteleros o similares. Este sistema está diseñado para:

* **Facilitar la gestión de habitaciones:** Utilizando estructuras de datos avanzadas como árboles, el sistema organiza y administra la información sobre las habitaciones disponibles, ocupadas y reservadas de manera eficiente.
* **Optimizar el proceso de reserva:** Empleando ArrayLists, el sistema gestiona las reservas realizadas, permitiendo agregar, modificar y eliminar reservas de forma rápida y sencilla.
* **Gestionar la disponibilidad de días:** A través de listas, el sistema mantiene un seguimiento preciso de los días disponibles para realizar reservas, asegurando que no haya conflictos ni sobreventas.
* **Proveer una interfaz intuitiva:** El sistema está diseñado para ser intuitivo y fácil de usar, minimizando el tiempo de capacitación necesario para que los usuarios puedan comenzar a utilizarlo de manera efectiva.
* **Mejorar la experiencia del usuario:** Al automatizar y simplificar el proceso de reserva, el sistema contribuye a mejorar la satisfacción tanto de los administradores como de los clientes, ofreciendo una experiencia de reservación más ágil y confiable.

La arquitectura del sistema de reservaciones está diseñada para ser modular y escalable, facilitando la gestión eficiente de las habitaciones, reservas y días disponibles. A continuación se presenta una visión general de los componentes principales y cómo interactúan entre sí.

**Componentes Principales**

**Gestión de Habitaciones (Árboles)**

**Estructura de Datos:** Utiliza una estructura de árbol binario para organizar las habitaciones. Cada nodo del árbol representa una habitación y contiene información relevante como el número de la habitación, su estado (disponible, reservada, ocupada), y otros detalles específicos (tipo de habitación, capacidad, etc.).

**Funcionalidades:** Permite agregar, eliminar y buscar habitaciones de manera eficiente. La estructura de árbol asegura un tiempo de búsqueda óptimo, lo que mejora el rendimiento del sistema.

**Gestión de Reservas (ArrayList)**

**Estructura de Datos:** Las reservas se gestionan utilizando un ArrayList, que almacena objetos de tipo Reserva. Cada objeto Reserva contiene detalles como el identificador de la reserva, el identificador de la habitación, la fecha de inicio y fin de la reserva, y los datos del cliente.

**Funcionalidades:** Permite agregar, modificar y eliminar reservas. El uso de ArrayList facilita la manipulación de las reservas y proporciona una estructura flexible para almacenar un número variable de elementos.

**Gestión de Días Disponibles (Listas)**

**Estructura de Datos:** Utiliza listas para gestionar los días disponibles para reservas. Cada elemento de la lista representa un día específico y contiene información sobre la disponibilidad de las habitaciones en esa fecha.

**Funcionalidades:** Permite agregar y actualizar la disponibilidad de días de manera dinámica. La estructura de listas facilita el seguimiento y la modificación de la disponibilidad a lo largo del tiempo.

**Interacción entre Componentes**

**Flujo de Datos**

**Reserva de una Habitación:** Cuando un usuario realiza una reserva, el sistema consulta el árbol de habitaciones para verificar la disponibilidad, luego actualiza el ArrayList de reservas con la nueva reserva y ajusta la lista de días disponibles.

**Modificación de una Reserva:** El sistema busca la reserva específica en el ArrayList, realiza las modificaciones necesarias y actualiza la disponibilidad en la lista de días correspondientes.

**Gestión de Disponibilidad:** La lista de días disponibles se actualiza dinámicamente en función de las reservas realizadas, garantizando que la información de disponibilidad sea precisa y actualizada en todo momento.

**Interfaz de Usuario**

La interfaz de usuario está diseñada para ser intuitiva, permitiendo a los usuarios interactuar fácilmente con el sistema para realizar reservaciones, consultar disponibilidad y gestionar habitaciones. Las acciones realizadas en la interfaz de usuario se traducen en operaciones sobre los componentes de datos mencionados anteriormente (árboles, ArrayList y listas).

**Clases**

**Clase Nodo Arbol**

**Figura 1**

*Clase nodo*

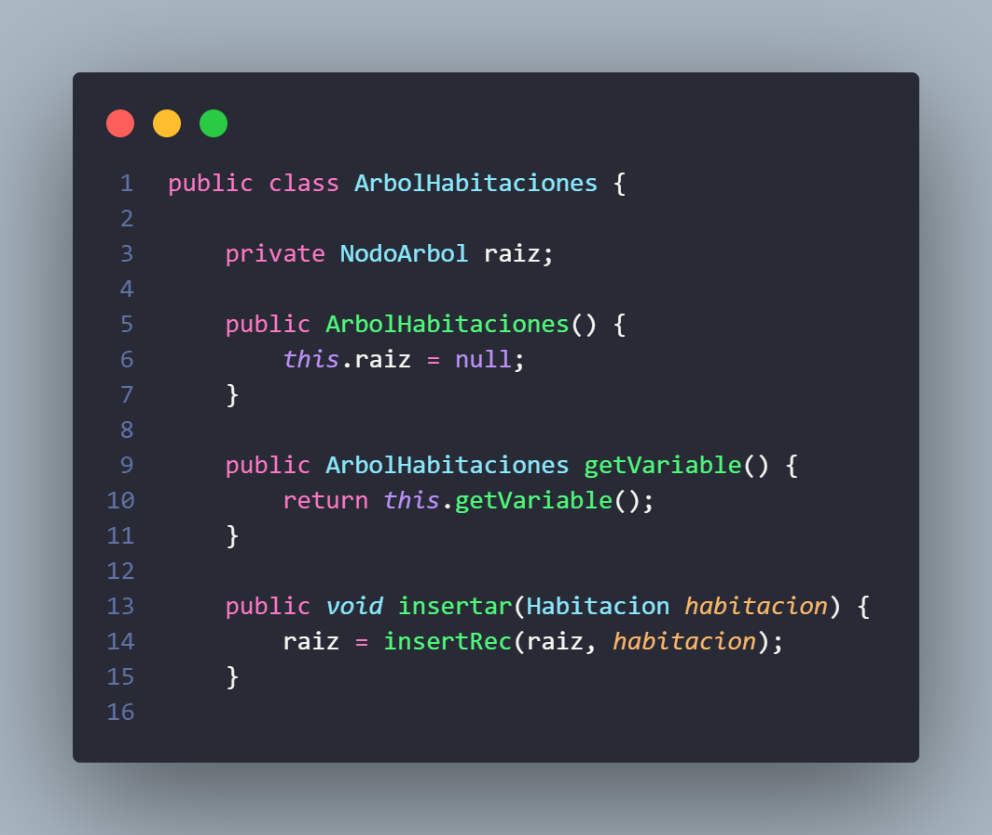


Esta clase nodo determina que información se guardara en cada espacio que se almacena dentro del árbol que simulara el sistema que gestiona las habitaciones.

**Árbol Habitaciones**

**Figura 2**

*Clase árbol habitaciones*

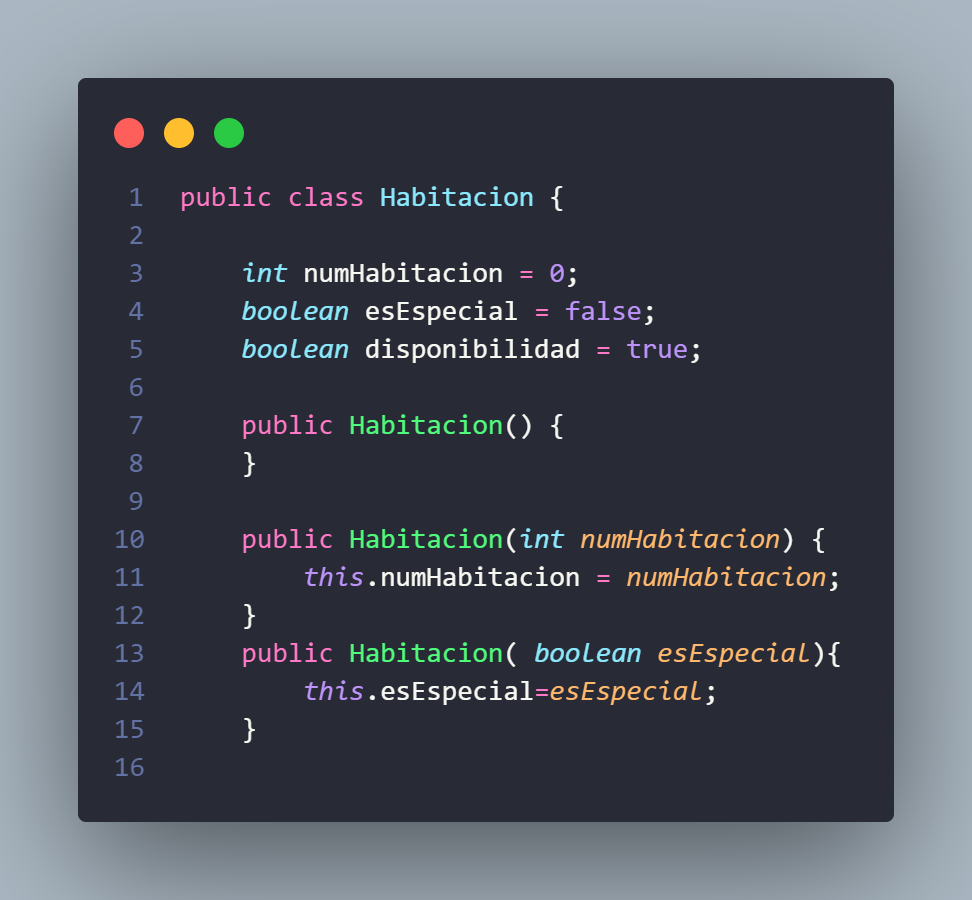


Esta estructura simula la estructura física y organizativa del hotel, simulando la disposición del espacio físico de un hotel y separando las habitaciones en tipos, especial y standard.

**Clase Habitación**

**Figura 3**

*Código clase Habitación*

****

Esta clase simula una habitación del hotel con sus diferentes características, como su numero o id, el atributo isEspecial que determina si es una clase especial o normal, y el ultimo disponibilidad, que determina si la habitación esta reservada.

**Clase Reserva**

**Figura 5**

*Código clase Reserva*

****

La clase reserva es donde se guardaran los datos que conectaran al usuario cliente con la habitación que desea reservar, de forma que se guarden los atributos que identifican al cliente, como lo son: nombre, nDocu (número de documento), edad, teléfono, tiempo de estadía, numero de personas que se alojaran en la habitación, y numero de habitación; el atributo final es un arreglo de tipo boolean que permite registrar los días que será reservada la habitación

**Clase Vista Principal**

**Figura 6**

*Codigo vista Principal*

Texto

Descripción generada automáticamente

Esta clase es aquella que permite visualizar el entorno grafico perteneciente a la ventana principal, donde el usuario se podrá dirigir a las otras ventanas, para reservar, eliminar reservas y obtener información del tipo de habitaciones, así como del hotel.

**Clase Vista info**

**Figura 7***Código vista ínfo*

**

Aquí el usuario podrá obtener una breve descripción del hotel y sus servicios

**Clase Vista habitaciones**

**Figura 8***Código Vista Habitaciones*

**Texto

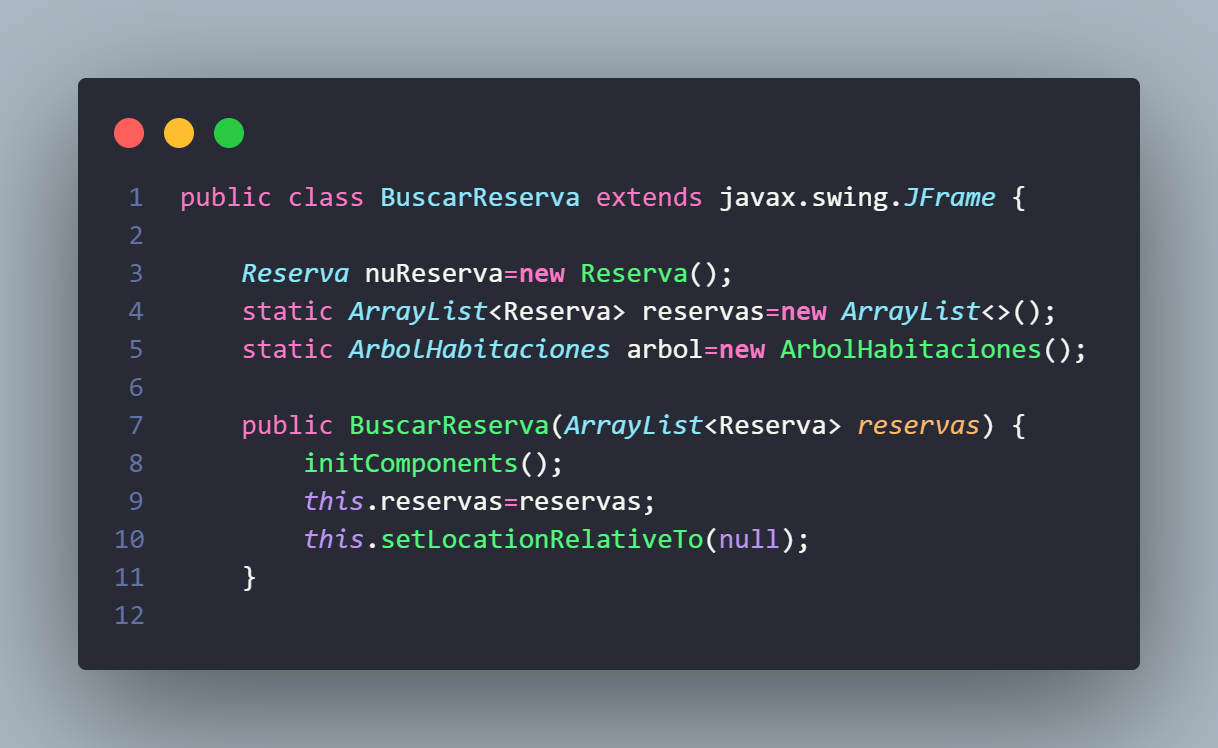
Descripción generada automáticamente**

Pantalla donde visualizar una descripción de las diferentes opciones de habitación disponibles.

**Clase Buscar Reserva**

**Figura 9**

*Código Buscar Reserva*

****

En esta clase se programo la funcionalidad de buscar reserva, donde el usuario podrá consultar las reservas que tenga guardadas a su nombre, y ver todos y cada uno de los detalles asociadas a las mismas.

**Clase Eliminar reserva**

**Figura 10**

*Código Eliminar Reserva*

****

Esta clase corresponde a la vista que nos dirige a la ventana correspondiente a eliminar reservas, donde el usuario podrá buscar la reserva que desea eliminar, desde que esta se encuentre almacenada en el sistema.

**Capacidades del sistema**

Actualmente el programa es un prototipo, que por el momento se maneja un máximo de 6 habitaciones, con posibilidad de aumentar, pero actualmente por razones de sostenibilidad del código se mantiene el límite en esos valores

* Es posible eliminar reservas existentes, pero no se pueden actualizar
* Es posible realizar reservas en un lapso corto de días.
* Es posible elegir entre dos tipos de habitaciones, y un máximo de 3 personas por habitación.
* No hace uso de bases de datos