**Documentación proyecto**

**Juan Camilo Mercado - 202021541**

**John Jairo Serrato Suarez – 202113080**

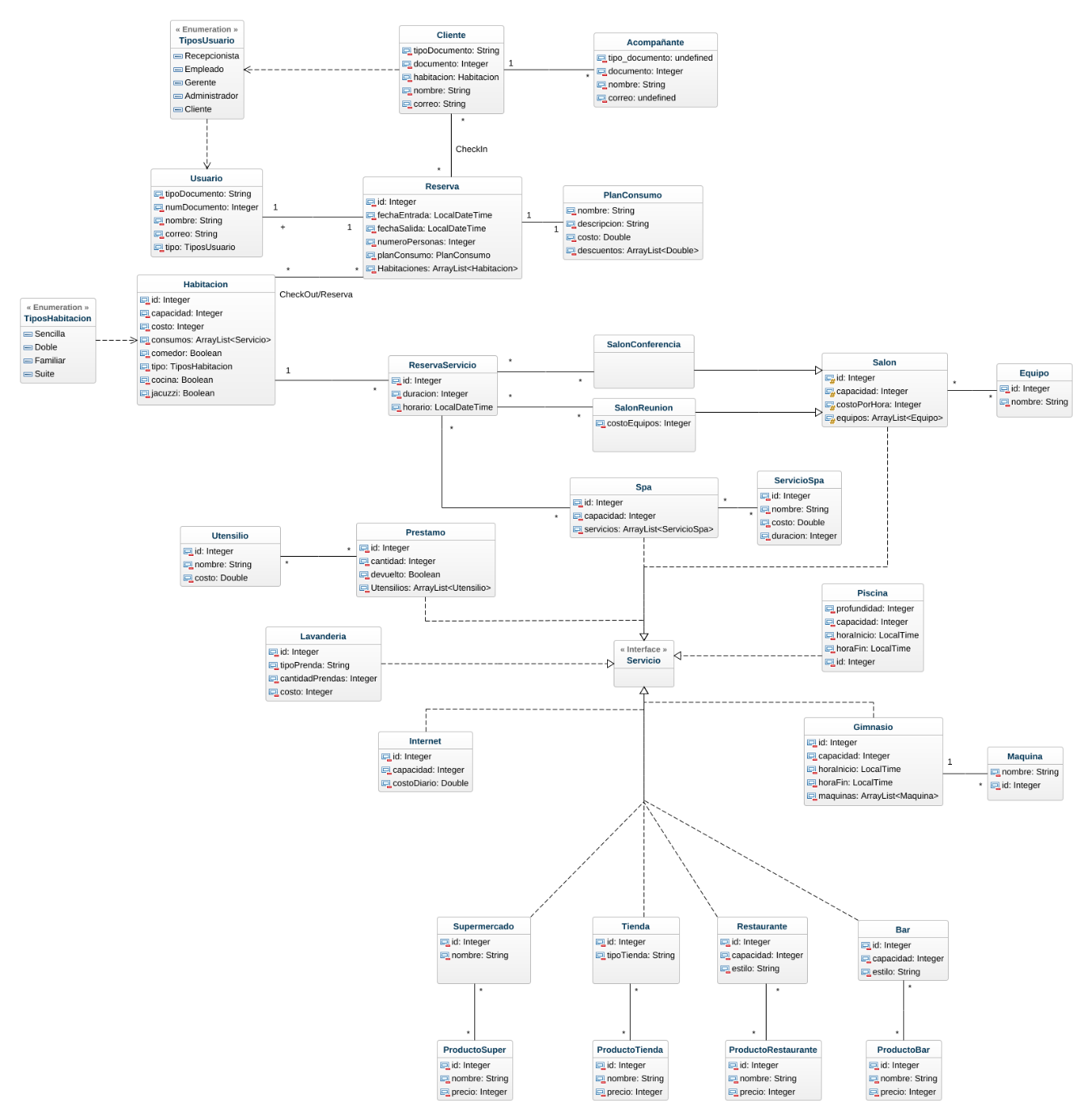
**Luis Borbon Holguin – 202110887**

**Sistemas transaccionales**

**Universidad de los Andes**

**01/Octubre/2023**

1. **Diagrama de clases**



El modelo E/R y relacional se encuentran en los documentos modelorelacional.pdf y modelo.dmd dentro de la careta modelo entidad relación. No se colocan en el presente documento por legibilidad. Incluso se recomienda abrir el Excel de modelo relacional en vez del pdf por claridad en las tablas.

1. **Requerimientos funcionales**

|  |  |
| --- | --- |
| Nombre | RF1. Registrar, actualizar y borrar tipos de usuario |
| Resumen | Se quiere que de los tipos de usuario definidos en la descripción del negocio |
| Entradas | |
| Información de los tipos de los usuarios, como Tipo de documento y numero de documento. | |
|  | |
|  | |
| Resultados | |
| El administrador de datos del hotel pude actualizar, registrar y borrar consultar el usuario como su identificación, nombre, correo electrónico y su rol respectivo. | |
|  | |

|  |  |
| --- | --- |
| Nombre | RF2. Registrar, actualizar y borrar consultar usuario |
| Resumen | Se quiere los usuarios quienes tienen una identificación (Tipo de documento, número de documento), un nombre, un correo electrónico y su rol respectivo. |
| Entradas | |
| Información de los usuarios, como Tipo de documento y numero de documento. | |
|  | |
|  | |
| Resultados | |
| El administrador de datos del hotel pude actualizar, registrar y borrar consultar el usuario como su identificación, nombre, correo electrónico y su rol respectivo. | |

|  |  |
| --- | --- |
| Nombre | RF2. Registrar, actualizar y borrar consultar usuario |
| Resumen | Se quiere los usuarios quienes tienen una identificación (Tipo de documento, número de documento), un nombre, un correo electrónico y su rol respectivo. |
| Entradas | |
| Información de los usuarios, como Tipo de documento y numero de documento. | |
|  | |
|  | |
| Resultados | |
| El administrador de datos del hotel pude actualizar, registrar y borrar consultar el usuario como su identificación, nombre, correo electrónico y su rol respectivo. | |
|  | |

|  |  |
| --- | --- |
| Nombre | RF3. Registrar, actualizar y borrar consultar tipo de habitación |
| Resumen | De los tipos de habitaciones de los cuales dispone el hotel con su dotación correspondiente. |
| Entradas | |
| Información de los usuarios, su tipo de habitación de los cuales dispone el hotel con su dotación correspondiente | |
|  | |
|  | |
| Resultados | |
| El administrador del hotel puede cambiar registrar, actualizar y borrar consultar tipo de habitación | |
|  | |

|  |  |
| --- | --- |
| Nombre | RF4. Registrar, actualizar y borrar consultar habitación |
| Resumen | De las instancias de las habitaciones de las cuales dispone el hotel |
| Entradas | |
| Información de las habitaciones con las que cuenta el hotel de los Andes y sus instancias. | |
| Resultados | |
| El administrador de datos del hotel puede registrar, actualizar y borrar la habitación que se quedan los huéspedes del hotel. | |
|  | |

|  |  |
| --- | --- |
| Nombre | RF5. Registrar, actualizar y borrar consultar un servicio del hotel |
| Resumen | De la descripción de los servicios de los cuales dispone el hotel. |
| Entradas | |
| La información de servicios que dispone el hotel | |
|  | |
|  | |
| Resultados | |
| El administrador de datos del hotel puede borrar, actualizar y borrar un servicio de hotel | |
|  | |

|  |  |
| --- | --- |
| Nombre | RF6. Registrar, actualizar y borrar consultar un plan de consumo |
| Resumen | De los planes de consumo disponibles en el hotel |
| Entradas | |
| Información de los planes de consumo | |
|  | |
|  | |
| Resultados | |
| El administrador de datos del hotel pude actualizar, registrar y borrar consultar el un plan de consumo | |
|  | |

|  |  |
| --- | --- |
| Nombre | RF7. Registrar, actualizar y borrar consultar una reserva de alojamiento |
| Resumen | De la reservación de una habitación, poder hacer cambios a esta |
| Entradas | |
| Información de la reserva del alojamiento | |
| Resultados | |
| El cliente con una reservación actualizar, registrar y borrar consultar el usuario como su periodo de tiempo, esto se puede hacer siempre y cuando esté disponible. | |
|  | |

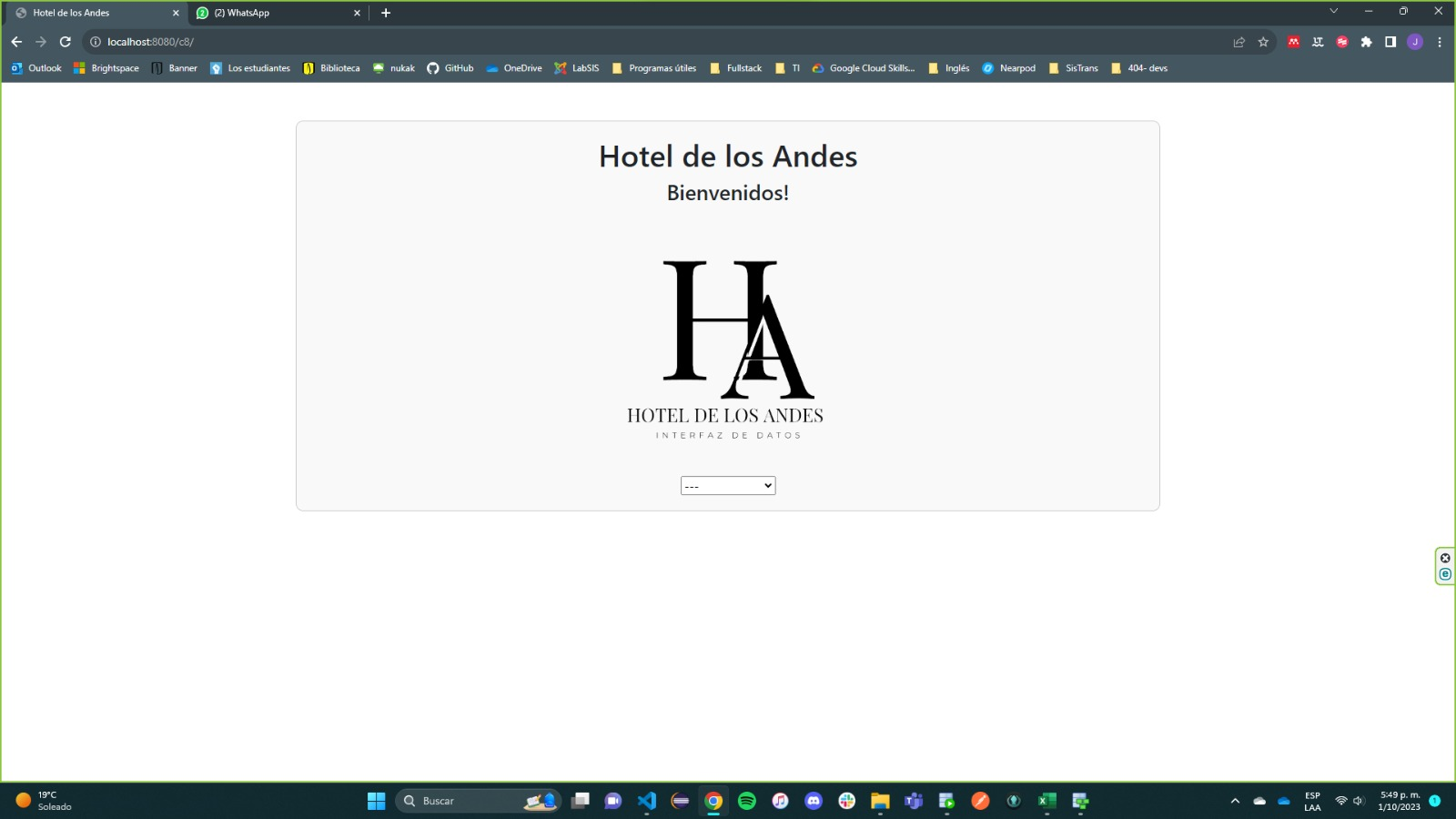
|  |  |
| --- | --- |
| Nombre | RF8. Registrar, actualizar y borrar consultar una reserva de un servicio del hotel por parte de un cliente |
| Resumen | De la reservación de un servicio por parte de un cliente |
| Entradas | |
| Información de la reserva de un servicio por parte de un cliente, un servicio se puede reservar siempre y cuando haya disponibilidad. | |
| Resultados | |
| El cliente puede cambiar registrar, actualizar una reserva de un servicio del hotel por parte de un cliente. | |

|  |  |
| --- | --- |
| Nombre | RF9. Registrar, actualizar y borrar consultar la llegada de un cliente al hotel |
| Resumen | De la llegada de un cliente de al hotel. |
| Entradas | |
| Información de los usuarios, la llegada de un cliente o sus acompañantes. | |
|  | |
|  | |
| Resultados | |
| El administrador de datos del hotel pude actualizar, registrar y borrar consultar el un plan de consumo | |
|  | |

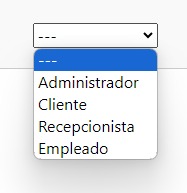
|  |  |
| --- | --- |
| Nombre | RF10. Registrar, actualizar y borrar consultar una reserva de alojamiento |
| Resumen | De la reservación de una habitación, poder hacer cambios a esta |
| Entradas | |
| Información de la reserva del alojamiento | |
| Resultados | |
| El cliente con una reservación actualizar, registrar y borrar consultar el usuario como su periodo de tiempo, esto se puede hacer siempre y cuando esté disponible. | |
|  | |

|  |  |
| --- | --- |
| Nombre | RF11. Registrar, actualizar y borrar consultar la salida de un cliente |
| Resumen | Del registro un consumo de la salida de un cliente al hotel, con todo lo que se implica. |
| Entradas | |
| Información de la reserva de un servicio por parte de un cliente, un servicio se puede reservar siempre y cuando haya disponibilidad. | |
|  | |
|  | |
| Resultados | |
| El cliente puede cambiar registrar, actualizar la salida de un cliente en el hotel y todo lo que implica por parte de un cliente. | |

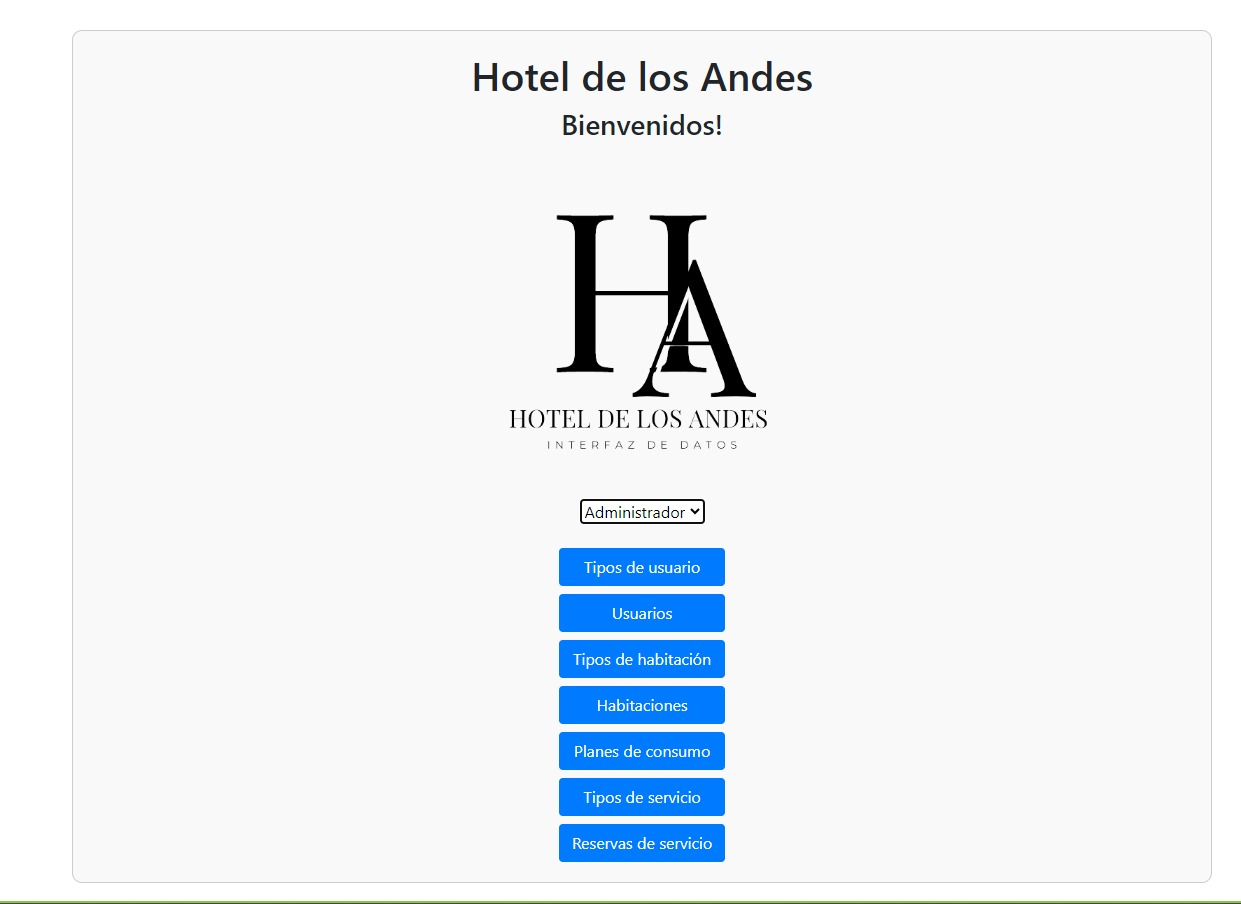
1. **Funcionamiento**



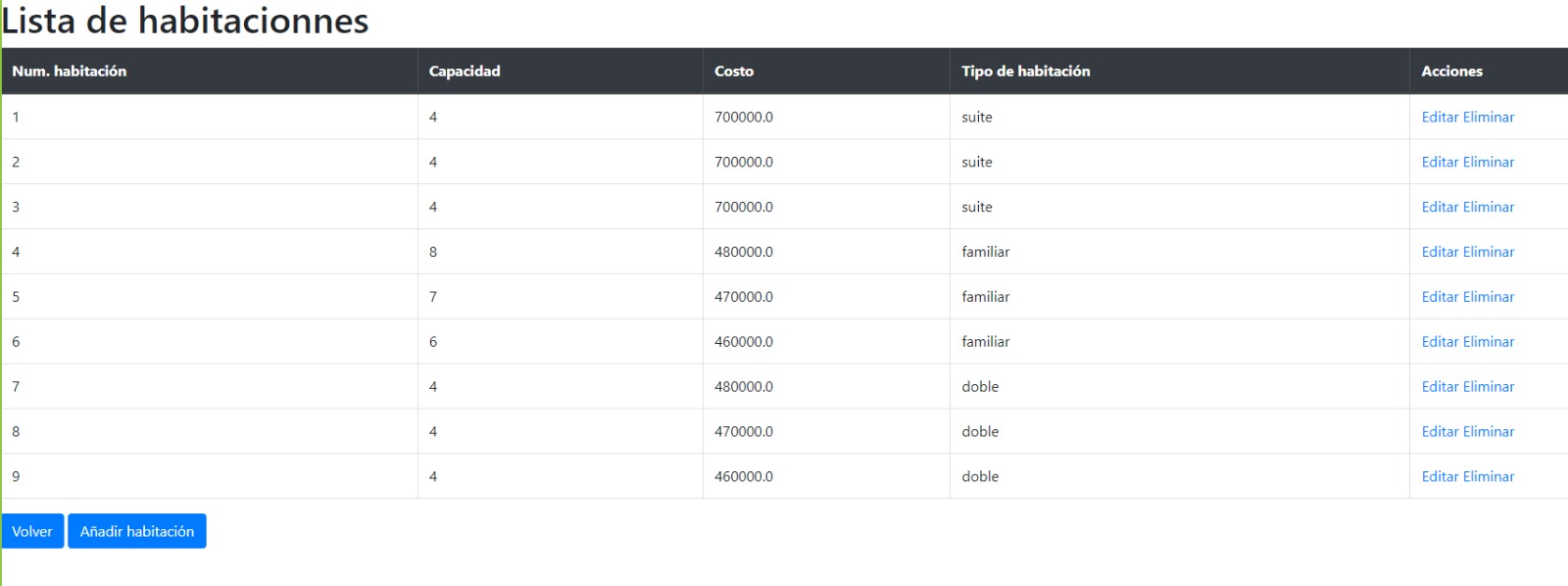
Al entrar a la pagina de inicio y está un menú desplegable que elige si es administrador de datos, cliente, recepcionista o empleado, según eso se despliegan las opciones correspondientes que son del enunciado.



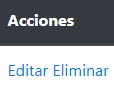
Luego de eso uno selecciona cualquier botón en verdad todos hacen las mismas cuatro acciones.



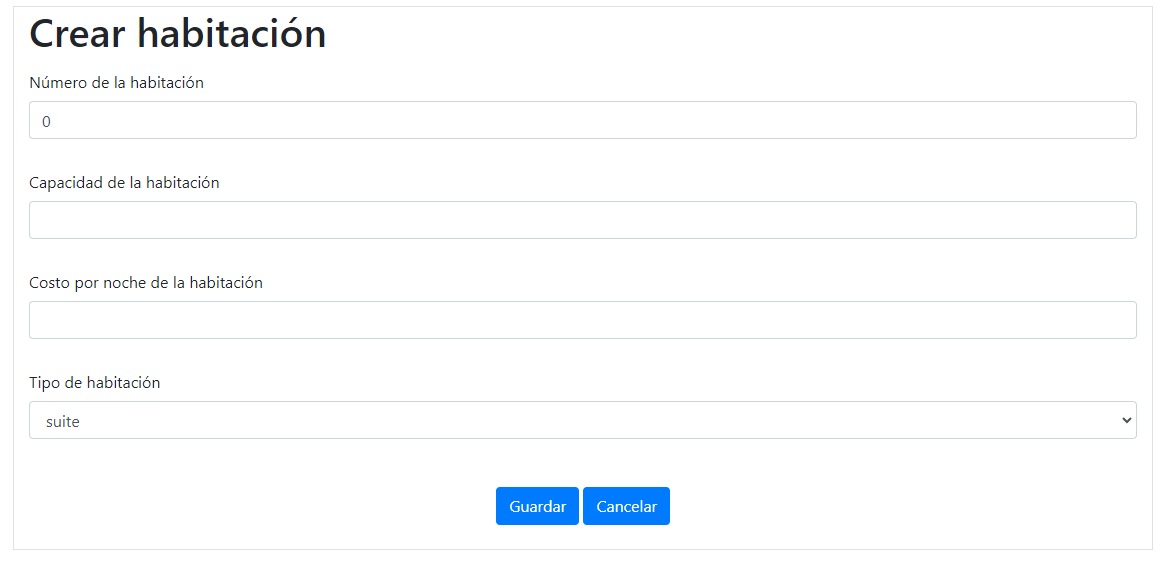
Al seleccionar el tipo de requerimiento aparece una tabla con todos los datos correspondientes de ese atributo, va a aparecer la opción de eliminar y editar, la tupla específica. En la parte inferior se encuentra el botón de volver y añadir habitación.



En la parte derecha aparece la columna de acciones, la cuales tienen accion de editar y eliminar.



Al seleccionar el botón añadir de habitación, se despliega un formulario, el cual piden los siguientes datos para la creación de datos.



Esta estructura es parecida para todos los usuarios de la pagina y todos los requerimientos requeridos para el proyecto.

**4. Supuestos y resultados**

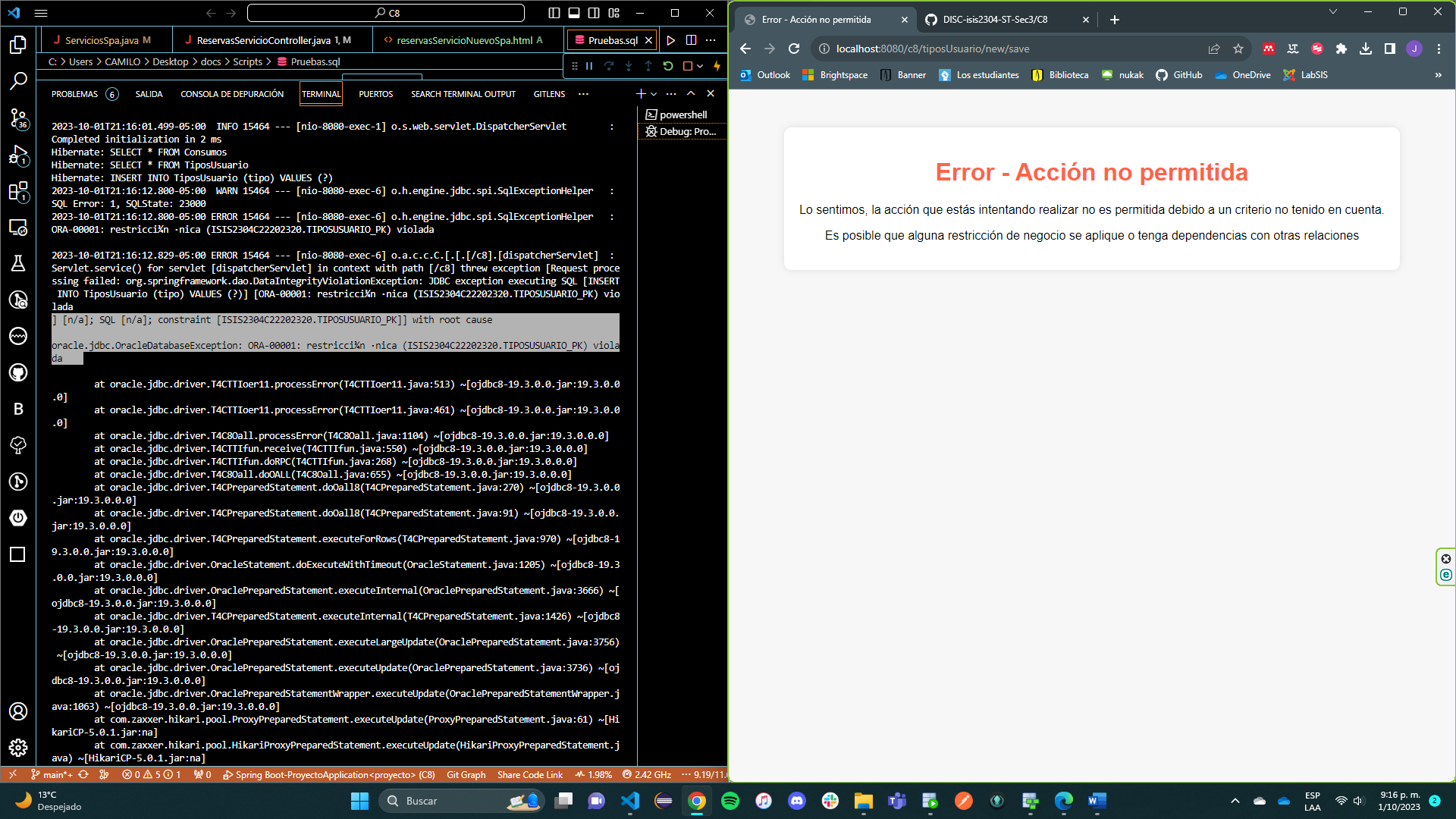
Los supuestos adicionales en las reglas del negocio se tomaron los check in y check out como operaciones mayormente humanas por lo tanto en la base de datos solo está el registro de estas sin las verificaciones completas.

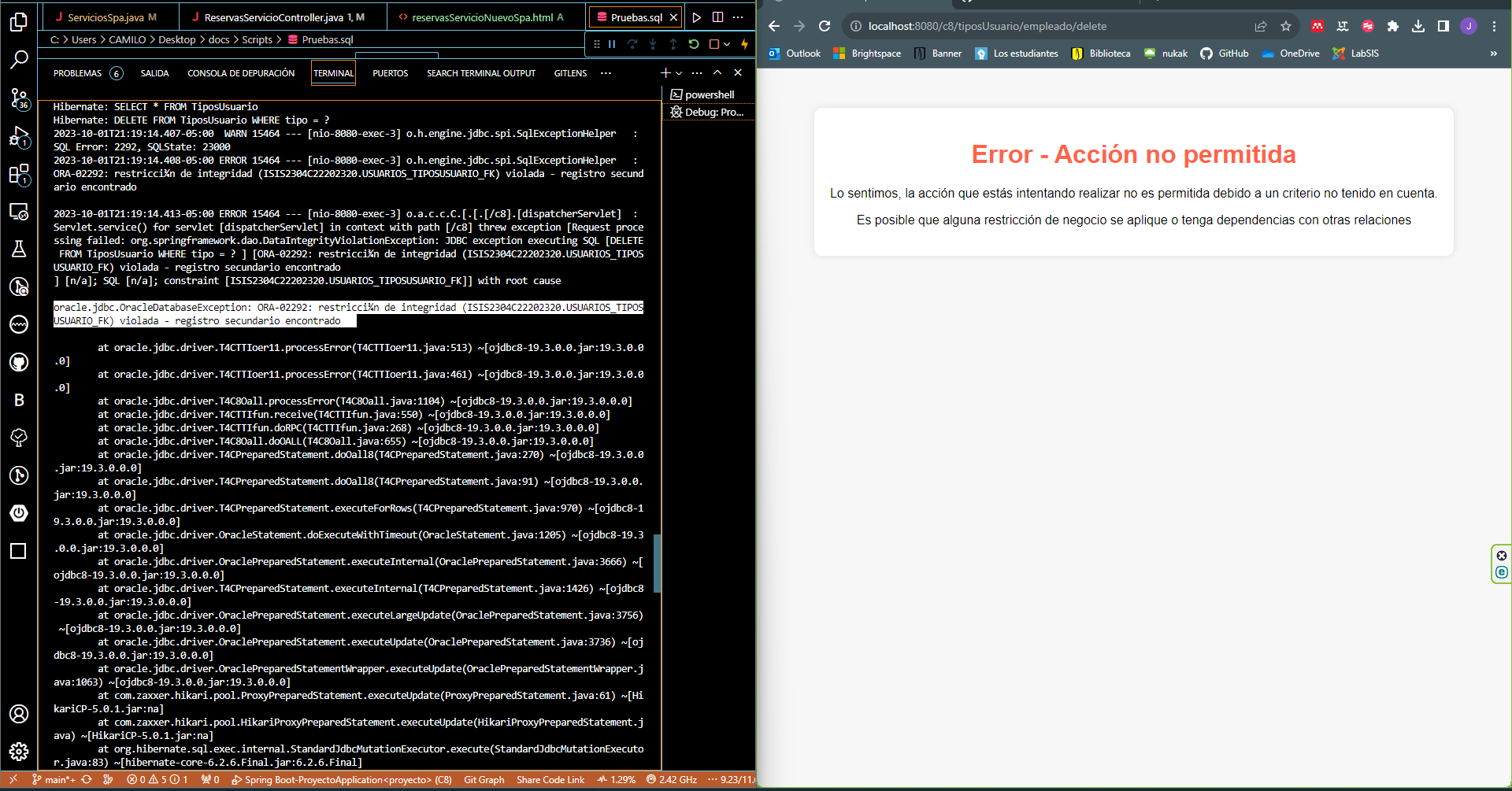
Respecto a los resultados logrados en su matoría los requerimientos se lograron implementar salvo el 9 que debido a un problema de la entidad check in que tiene una llave embebida dentro de otra llave embebida es complejo armar la entidad y spring no encuentra una manera de convertir de la base de datos a la entidad. Por otra parte, en el requerimiento 10 se tuvo una dificultad en la implementación y por falta de tiempo no se logró implementar el resultado esperado. En general se logra implementar correctamente los CRUD, en ocasiones la operación update genera problemas al estar vinculado como llave foránea a otras relaciones y genera un error al tratar de editarlo. Igualmente, se reconoce que hubo dificultades en la verificación de restricciones para poder o no hacer las operaciones de la B. La principal dificultad fue la implementación de las consultas complejas. Por otra parte, la comunicación fue un inconveniente dentro del desarrollo del proyecto, por el cual hubo dificultad en la implementación de los requerimientos.

El balance de las pruebas es correcto en cuanto se ejecutan las pruebas pedidas que se adjuntan en el archivo pruebas.sql en la carpeta scripts dentro de docs.

1. **Resultados de pruebas**

Para las pruebas de unicidad de tuplas con todos los casos puestos en el archivo pruebas sql en consola de spring se ven así, los errores se verifican correctos en la parte subrayada de la consola. Todos los mensajes de error son iguales para este caso.



Para las pruebas de integridad se ejecutan pruebas donde los datos tienen dependientes:  
  


En este caso se verifica que la excepción generada es correcta.

Por otra parte, en las pruebas de llave foránea no es posible hacerlas desde la interfaz de usuario ya que se diseñó para que solo se pudieran escoger llaves foráneas con menús desplegables y datos dependientes existentes. A pesar de esto se prueba en SQL developer con los casos de prueba de pruebas.sql y todos tienen salidas de la siguiente forma:  
  
Texto

Descripción generada automáticamente con confianza media

Viendo y verificando que las llaves foráneas se vinculan correctamente.

Por último las pruebas de integridad de chequeo obtuvieron el siguiente resultado en consola:



Donde la parte subrayada de la consola muestra el error de violación en la restricción del check. Este mismo mensaje de erro aparece para cada una de las pruebas realizadas en estos casos.

Siendo así, probados todos los casos de prueba para los diferentes escenarios se concluye que el diseño implementado maneja de manera correcta las reglas de negocio establecidas mediante restricciones en las relaciones.