DOCUMENTACIÓN APPRESERVAS

CARLOS FAROUK ABDALA RINCÓN
JUAN DIEGO GARCIA ESCOBAR
CESAR FELIPE GIRALDO MORA

- **Requisitos Funcionales:** Los requerimientos funcionales del aplicativo para los clientes son:
    - Inscribirse como nuevo cliente. (1)
    - Verificar la disponibilidad de los empleados con base a la tarea que se requiera, la franja horaria y el día. (2)
    - Realizar reservas con el empleado y para el servicio que desees por medio del aplicativo. (1)
    - Actualizar las citas previamente creadas. (3)
    - Cancelar las citas. (4)
    - Ver únicamente las citas del cliente. (2)

  Los requerimientos funcionales del aplicativo para las empresas o proveedores de los servicios son:
    - Inscribir empleados vinculados a un servicio o tarea y a una franja horaria que representara su disponibilidad. (1)
    - Crear franja horaria dependiendo de las características del negocio. (1)
    - Crear nuevas tareas para asignar empleados a las mismas. (1)
    - Ver todos los empleados, franjas horarias y tareas. (2)
    - Ver todas las citas reservadas con todos los detalles. (2)
    - Ver todos los clientes inscritos. (2)

  La manera de realizar todos estos requisitos funcionales fueron las siguientes:
    1. Haciendo uso de la etiqueta de Spring denominada Post Mapping en los controladores, se solicita la información en forma de Request Body de los Data Transfer Object deseados para crear o escribir un nuevo registro en la base de datos dependiendo de qué objeto se necesite crear (Task, Branch, Employee, Customer y Appointment) Para el caso de Appointment se pasa por una validación para verificar la disponibilidad antes de crear la cita. Seguido, se guarda la información en la base de datos haciendo uso de las interfaces Repository de cada objeto teniendo en cuenta la estructura que se le dio al momento de crear las respectivas clases.
    2. Para todas las funciones enfocadas a controlar o leer la base de datos se usa la etiqueta Spring Get Mapping, haciendo uso además de Request Param para consultar la disponibilidad de las citas y de Path Variable para buscar citas de un solo cliente. Se hace uso de la función de buscar implementada en los repositorios con JPA y se adaptan dependiendo el tipo de búsqueda.
    3. Para actualizar se hace uso de la etiqueta Spring Put Mapping que de manera similar al punto (1), hace uso de Request Body solicitando el DTO del objeto que se quiera trabajar y guarda haciendo uso del repositorio.

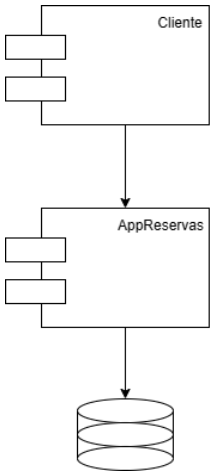
4. Para eliminar registros, se usa la etiqueta de Spring llamada Delete Mapping en la que nuevamente se solicita la información de la cita por medio de un DTO para que esta sea eliminada.

Estos requisitos son necesarios para poder asegurar el funcionamiento completo del sistema que se desea implementar. Cada requerimiento cumple una función especial para que los usuarios puedan hacer uso del sistema de la mejor manera y que puedan manejar a su conveniencia el mismo, ya sean empresas o clientes.
Es necesario que se pueda llevar un control de la información que se maneja, por ello, existen las funciones para ver la información.

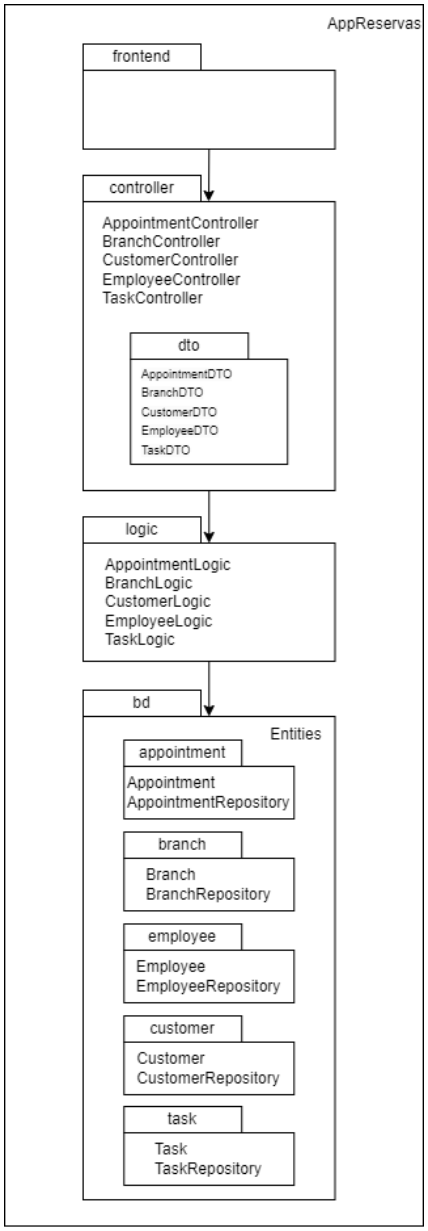
- **Requisitos No Funcionales:** El aplicativo cuenta principalmente con tres requisitos no funcionales que son los siguientes:
    - **Seguridad:** El aplicativo cuenta con un sistema de autenticación "Basic Authentication" que permite que solo usuarios autorizados puedan acceder a él, influyendo directamente en la seguridad del sistema. Sin las credenciales necesarias no se puede hacer uso del mismo.
    - **Usabilidad:** El sistema cuenta con una interfaz que facilita la manera en la que los clientes y empresas interactúan con el sistema, mejorando la forma en la que el usuario hace uso del aplicativo.
    - **Mantenibilidad:** La manera en la que se estructura el código permite que este sea mantenible a lo largo del tiempo, haciendo sencilla la gestión del mismo. Además se hace uso de herramientas como Spring que facilitan la implementación de este requerimiento. Para apoyar este requerimiento se implementaron inversiones de control y de dependencias para mejorar la estructuración del código y facilitar futuros cambios en la arquitectura de este mismo.

  Cada uno de estos requerimientos asegura que la experiencia de los usuarios al usar el aplicativo sea la mejor, y que el código pueda ser mantenible y escalable a lo largo del tiempo, evitando problemas que si bien, no son totalmente evidentes, pueden perjudicar significativamente la manera de interactuar con el sistema. Por lo anterior es importante velar por que cada uno de los requerimientos se mantenga vigente con los avances del aplicativo.
- **Estilo Arquitectónico:** El estilo arquitectónico usado en el aplicativo "AppReservas" es un estilo monolítico, pues todas las partes que se encargan del funcionamiento completo del aplicativo están en un mismo proyecto y hacen parte de un mismo componente como unidad lógica que puede funcionar de manera independiente.
  Si bien el proyecto del front-end se encuentra en un proyecto aparte, este es solo una manera de acceder o de facilitar la interacción con la aplicación , pero lo que es en sí el sistema se encuentra en un solo proyecto.
- **Patrones Arquitectónicos:**
    - **Arquitectura MVC:** El código se encuentra dividido siguiendo lo planteado por este patrón, pues la vista o interfaz gráfica se encuentra separada de los controladores y a su vez, ambos se separan de la lógica. De esta manera se asegura que cada una de estas partes trabaje de manera independiente y se facilita la modificación y manutención del código.

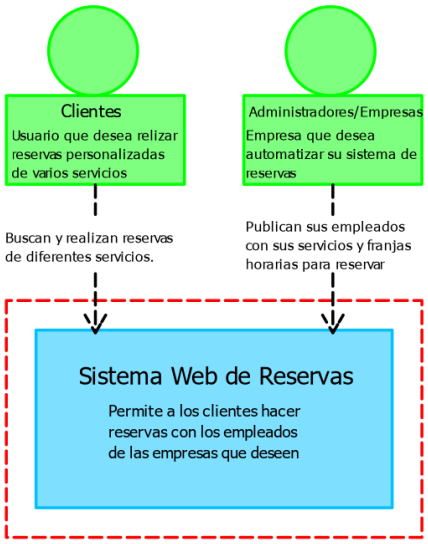
- **Diagrama 4+1:**
  - **Diagrama de Componentes:**



  - **Diagrama de Paquetes:**



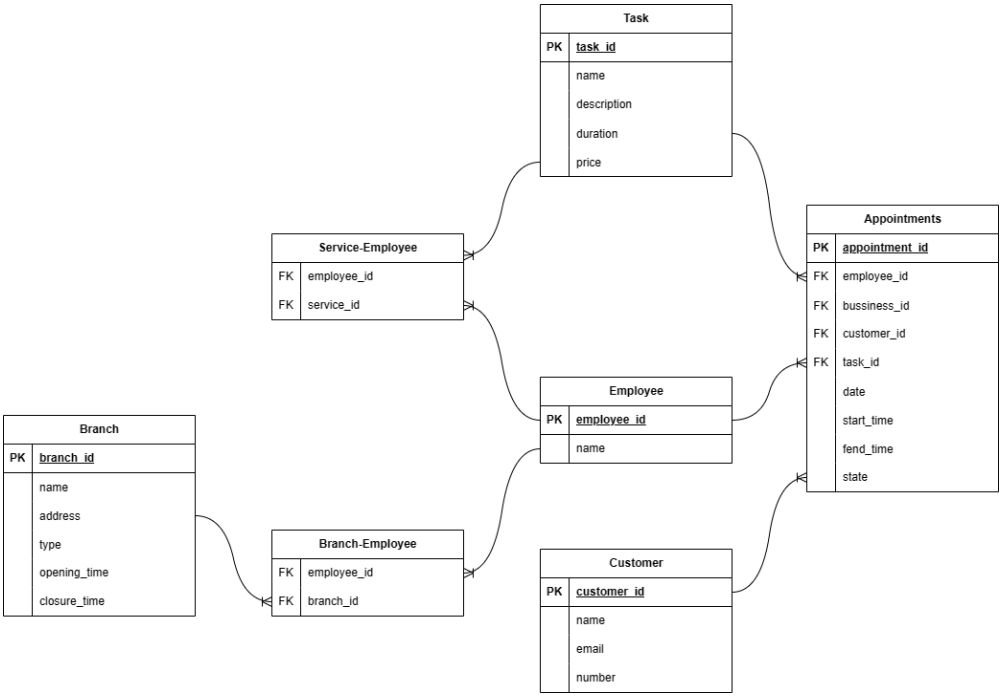
- **Diagramas C4:**
  - **Diagrama de Contexto del Sistema:**



  - **Diagrama de Contenedores:**

Diagrama de Contenedores para Sistema Web de Reservas

- **Diagramas Adicionales:**
  - **Diagrama Entidad-Relación:**



- **Responsabilidades Éticas:** El uso de un aplicativo de este tipo conlleva a tener que responder a muchas responsabilidades tanto éticas como profesionales.
    - El sistema debe ser transparente, cumplir con todas las normativas requeridas y no hacer ningún tipo de discriminación para clientes o empresas.
    - Debe velar por la legalidad a la hora de hablar de financiamiento, no se debe presentar ningún problema de este tipo por parte ni de los clientes ni de las empresas ni de los encargados del aplicativo.
    - Debe tener una responsabilidad social y ser diseñado para beneficiar a sus usuarios y no perjudicarlos, no se puede sacar provecho de manera fraudulenta del aplicativo ni de sus clientes.
    - La protección de los datos es fundamental, por ello como profesionales debemos velar por que estos nunca sean compartidos de manera indebida implementando sistemas apropiados para protegerlos.
    - Se debe además proteger al usuario y a las empresas, evitando que usuarios no autorizados puedan interactuar en el sistema.
    - Se debe mantener un registro que nos permita asegurar la fiabilidad del sistema en temas de cumplimiento para no perjudicar ni a clientes ni a empresas.
    - Se debe comprobar tanto la legalidad como la naturaleza de las empresas, pues es negativo para la sociedad el promover la práctica de actividades indebidas.
    - Debe ser imparcial y no velar por intereses individuales de ningún usuario ya sea cliente, empresa o algún colectivo o usuario externo.