|  |
| --- |
| **SPRINT 2: Desarrollo del Backend** |

|  |  |
| --- | --- |
| **Identificación Proyecto** | |
| Nombre Proyecto: | Desarrollo de una aplicación web para la gestión de citas de una peluquería |
| Número Equipo: | Equipo 7 |
| **Integrantes del equipo** | |
| Rol  (Líder-Desarrollador – Cliente) | Nombre |
| Líder | Christian Arias García |
| Desarrollador | Yessica Higuita Osorio |
| Desarrollador | Juan Felipe Gallego |
| Desarrollador | Álvaro Andrés Méndez |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| **Evidencia construcción del Backend** |

Creación del API REST utilizando Java + Spring Boot Como entorno de desarrollo utilizare Visual Studio Code Para configurar VSC al proyecto que voy a desarrollar instalare las siguientes extensiones: Extension

Pack for Java:IntelliJ IDEA – the Leading Java and - JetBrains

<https://www.jetbrains.com/idea/download/>

https://marketplace.visualstudio.com/items?itemName=vscjava.vscode-java-pack Spring Boot Extension Pack: <https://marketplace.visualstudio.com/items>?

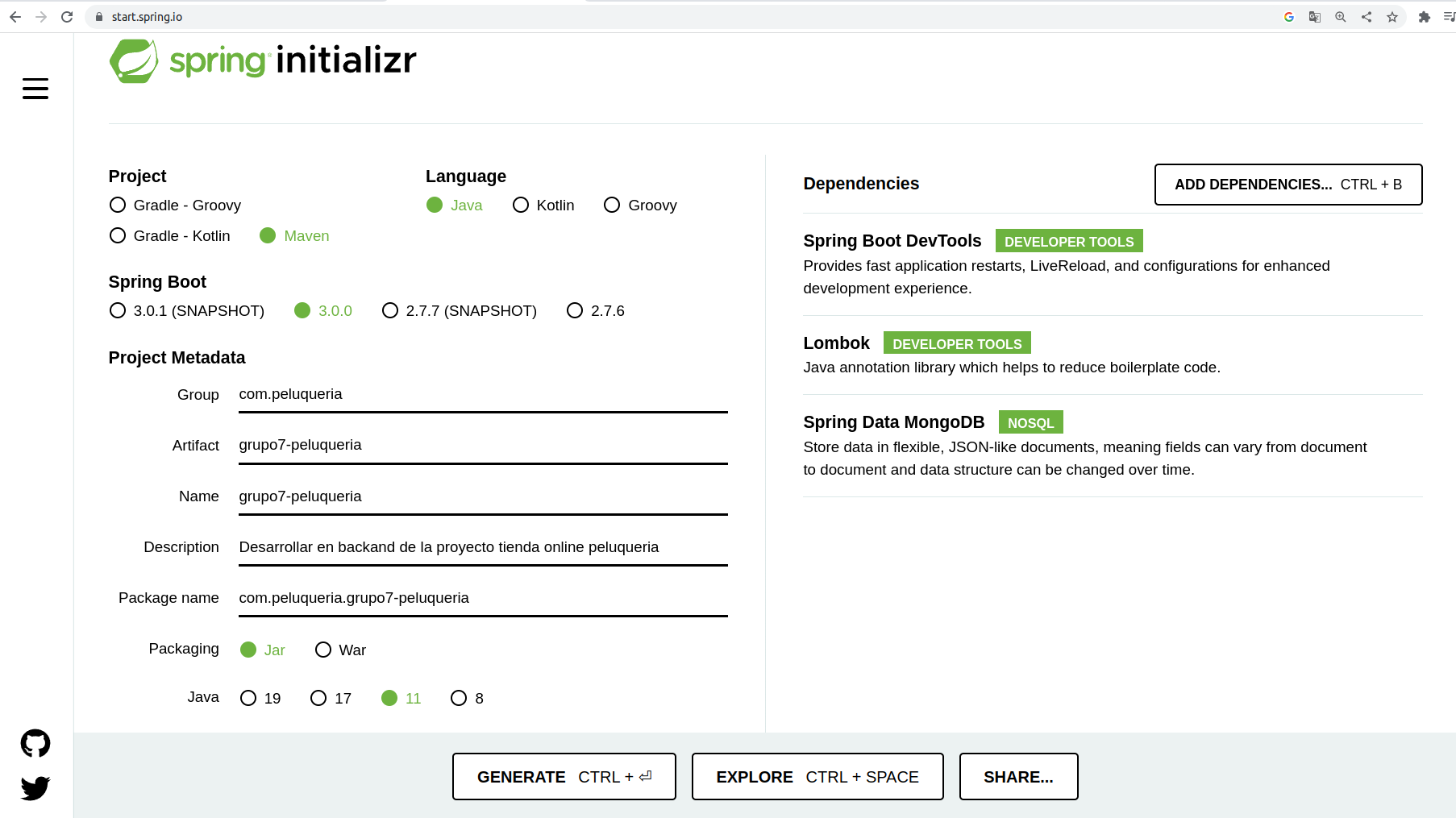
itemName=Pivotal.vscode-boot-dev-pack Como es un proyecto Java necesitamos tener instalado el JDK. A Java Development Kit (JDK):

https://adoptium.net/es/temurin/releases/?version=11 Para el sistema de base de datos vamos a utilizar MongoDB, que es un sistema basado en documentos. Utilizaremos

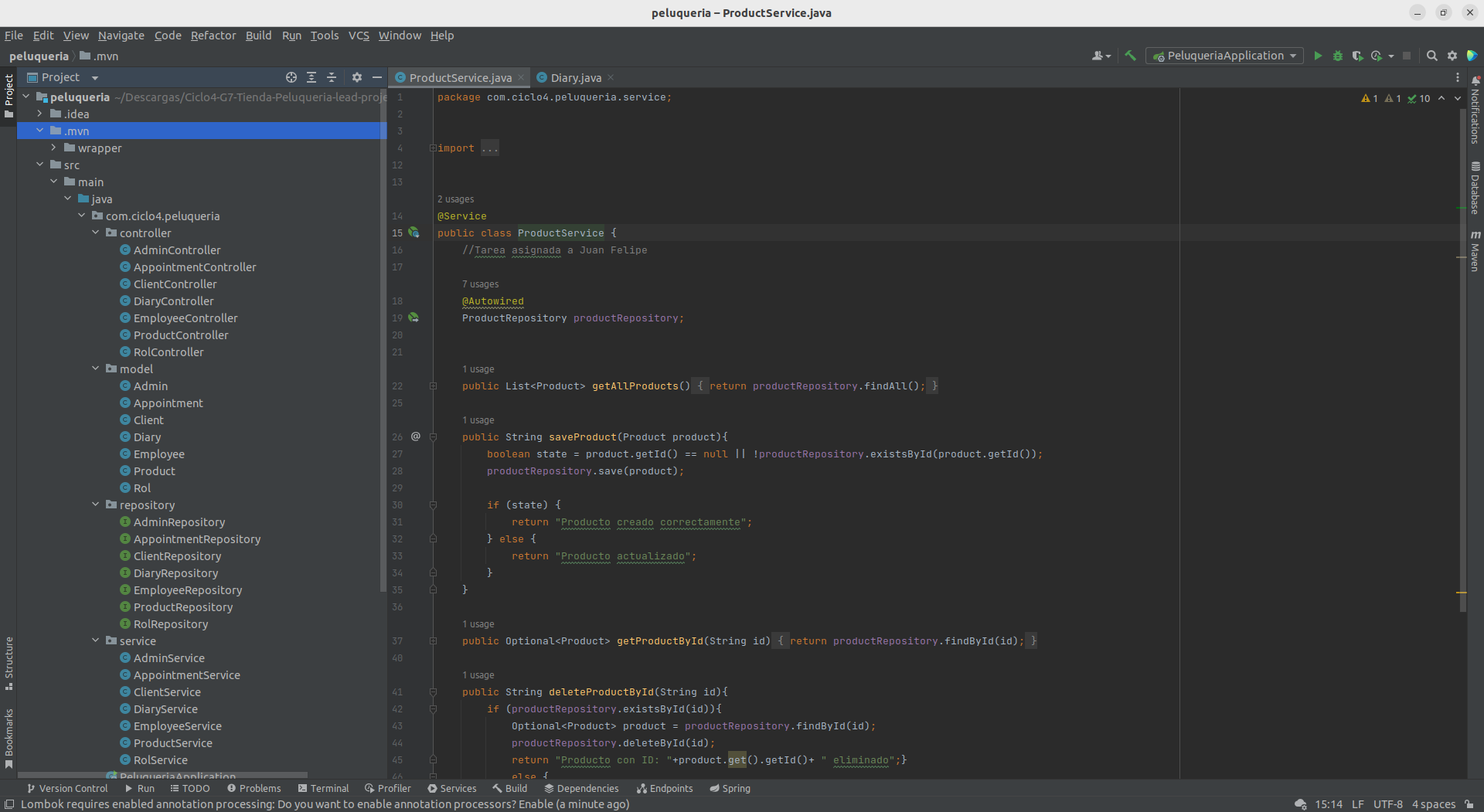
MongoDB Community Server: https://www.mongodb.com/try/download/community

Creamos el proyecto base del API REST: utilizando

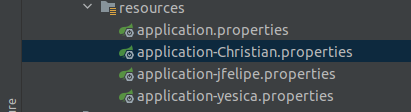
<https://start.spring.io/>



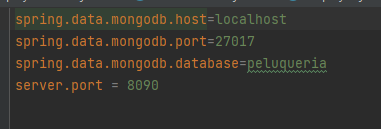
Creamos las entidades Client, Employee, Appoinment,Admin,Diary y Product con sus controladores, modelos,repositorio y servico

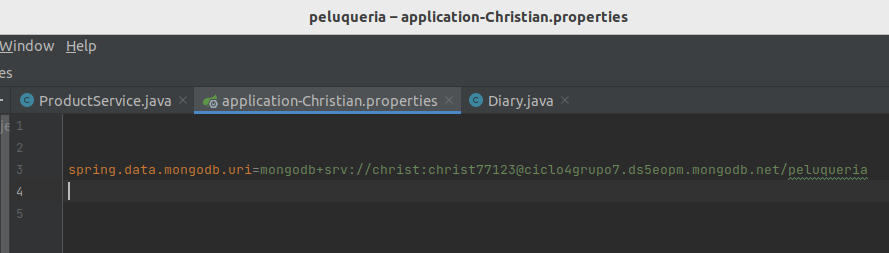


perfiles properties

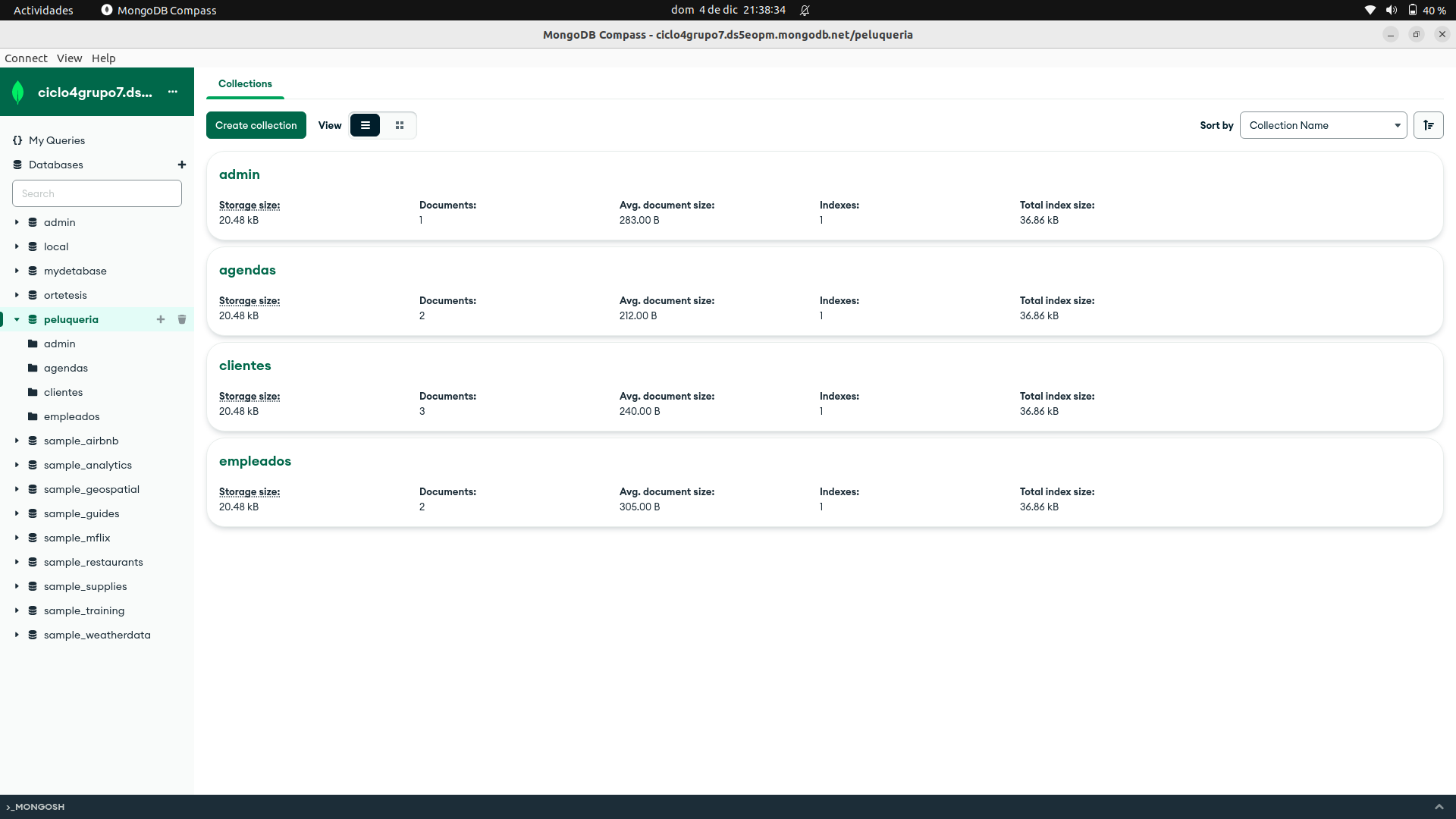


Conexión con base de datos

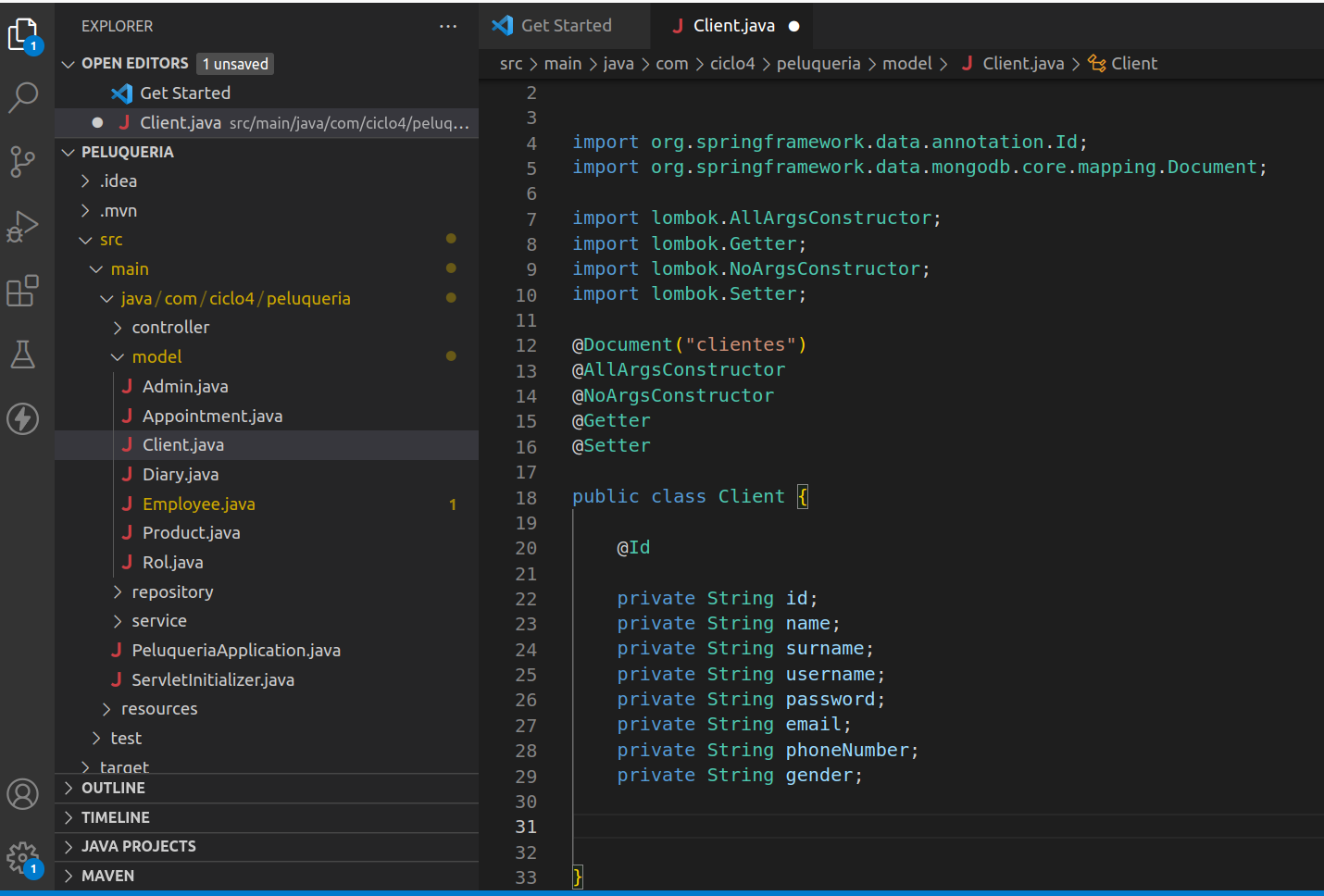




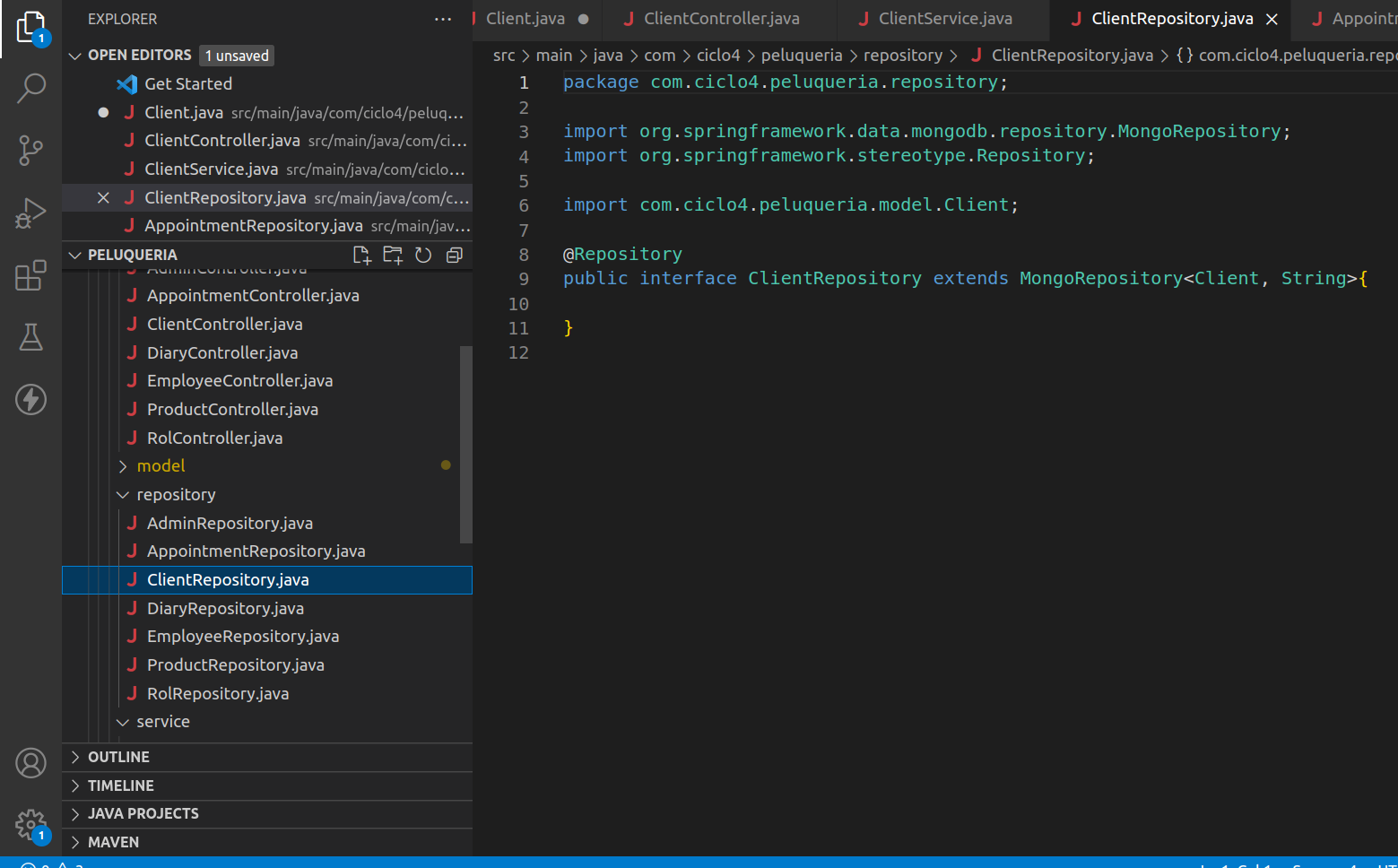
Base datos



Procedemos con la creación del modelo de Client:



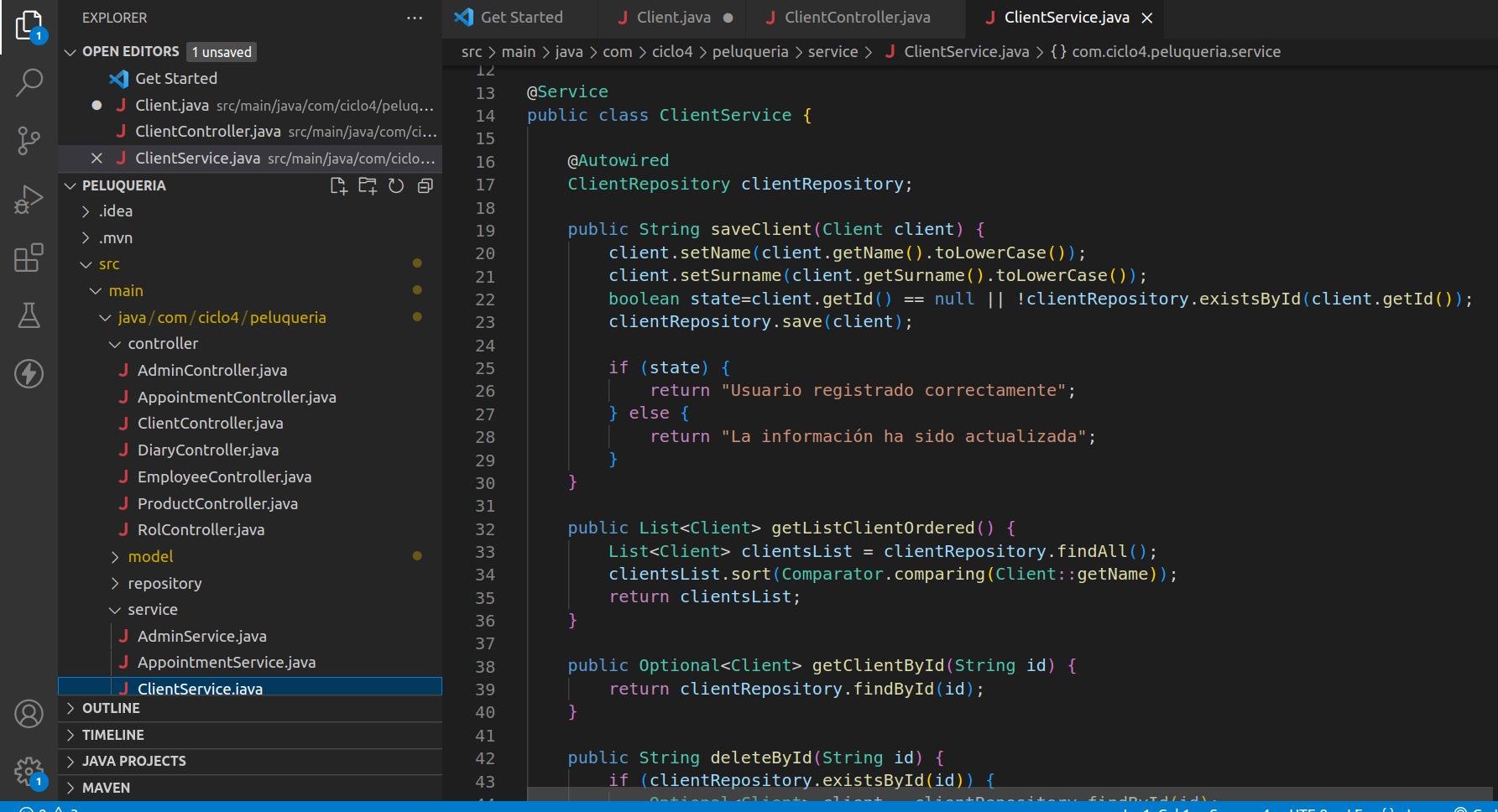
Creamos el repositorio (métodos CRUD) para el Client:



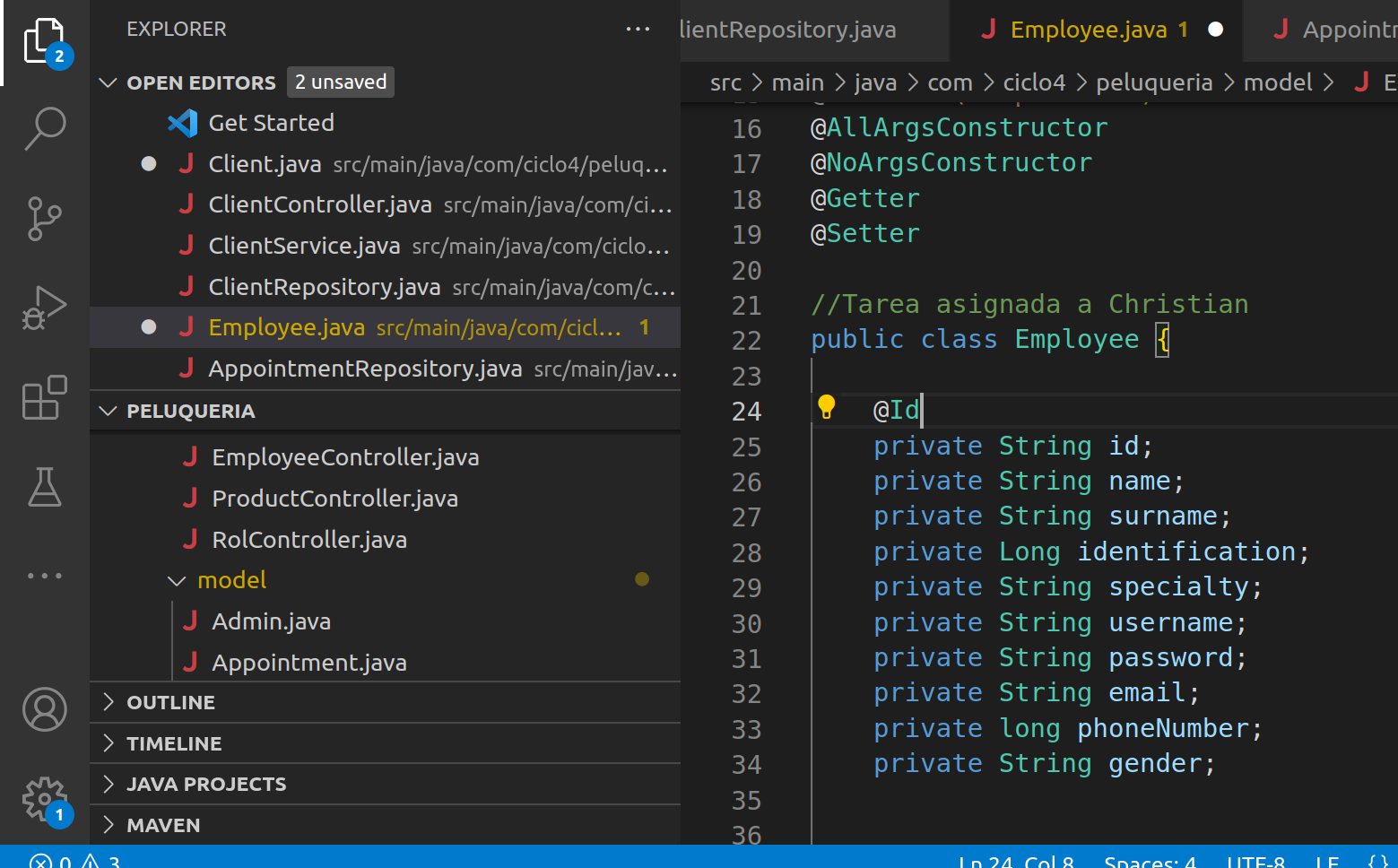
Crear el controlador Client:

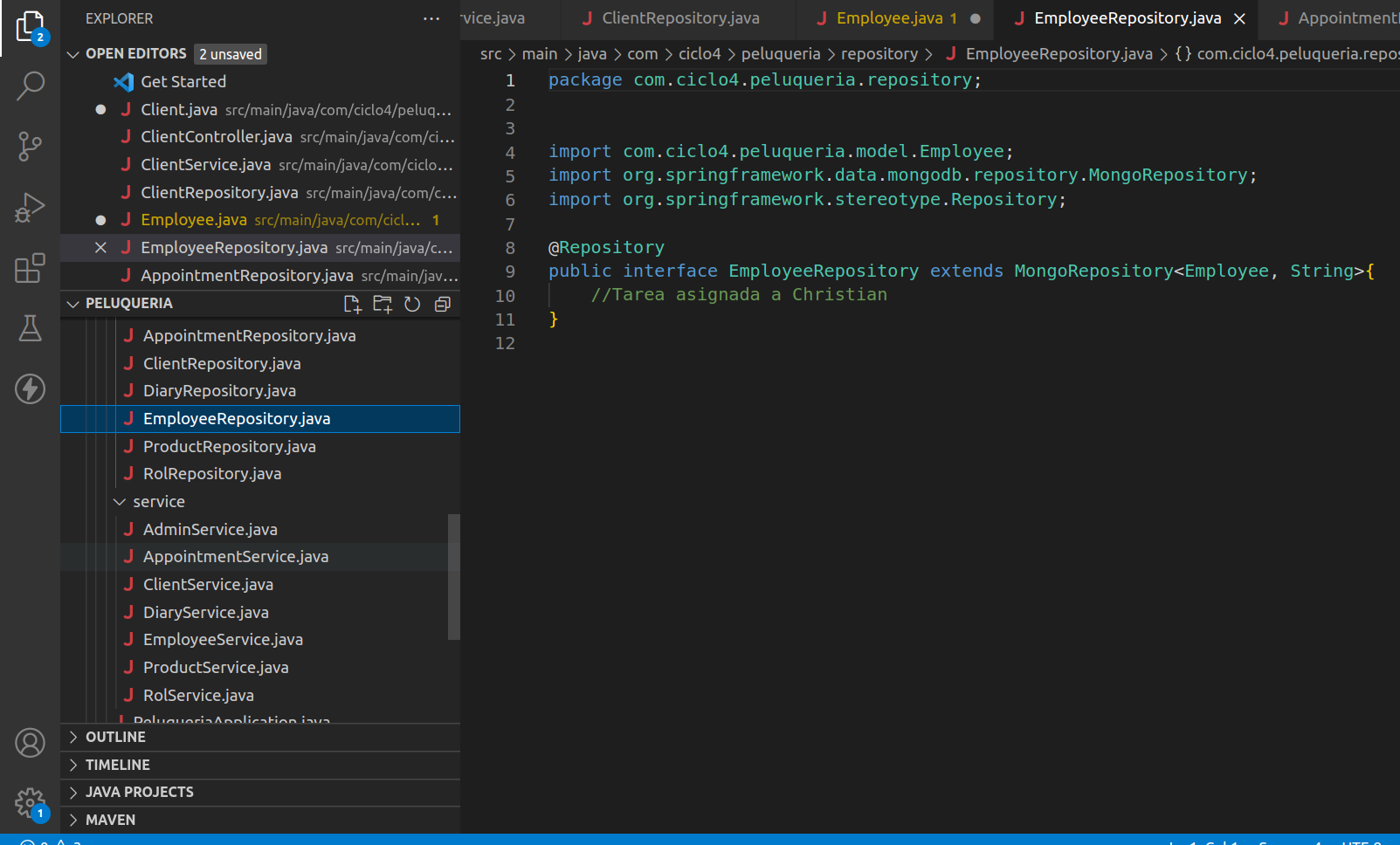


Creacion de servico Client:

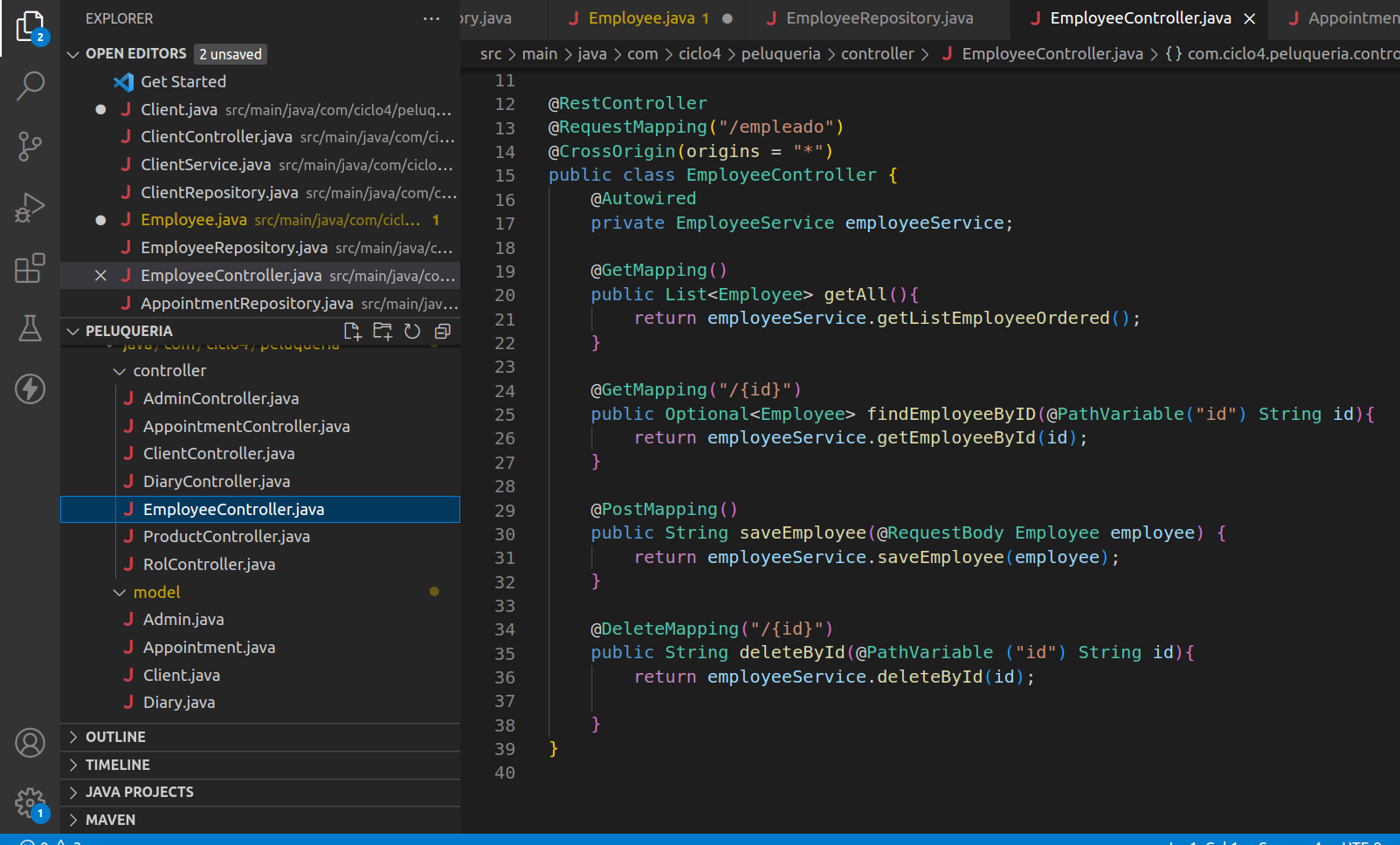


Procedemos con la creación del modelo de Employee:

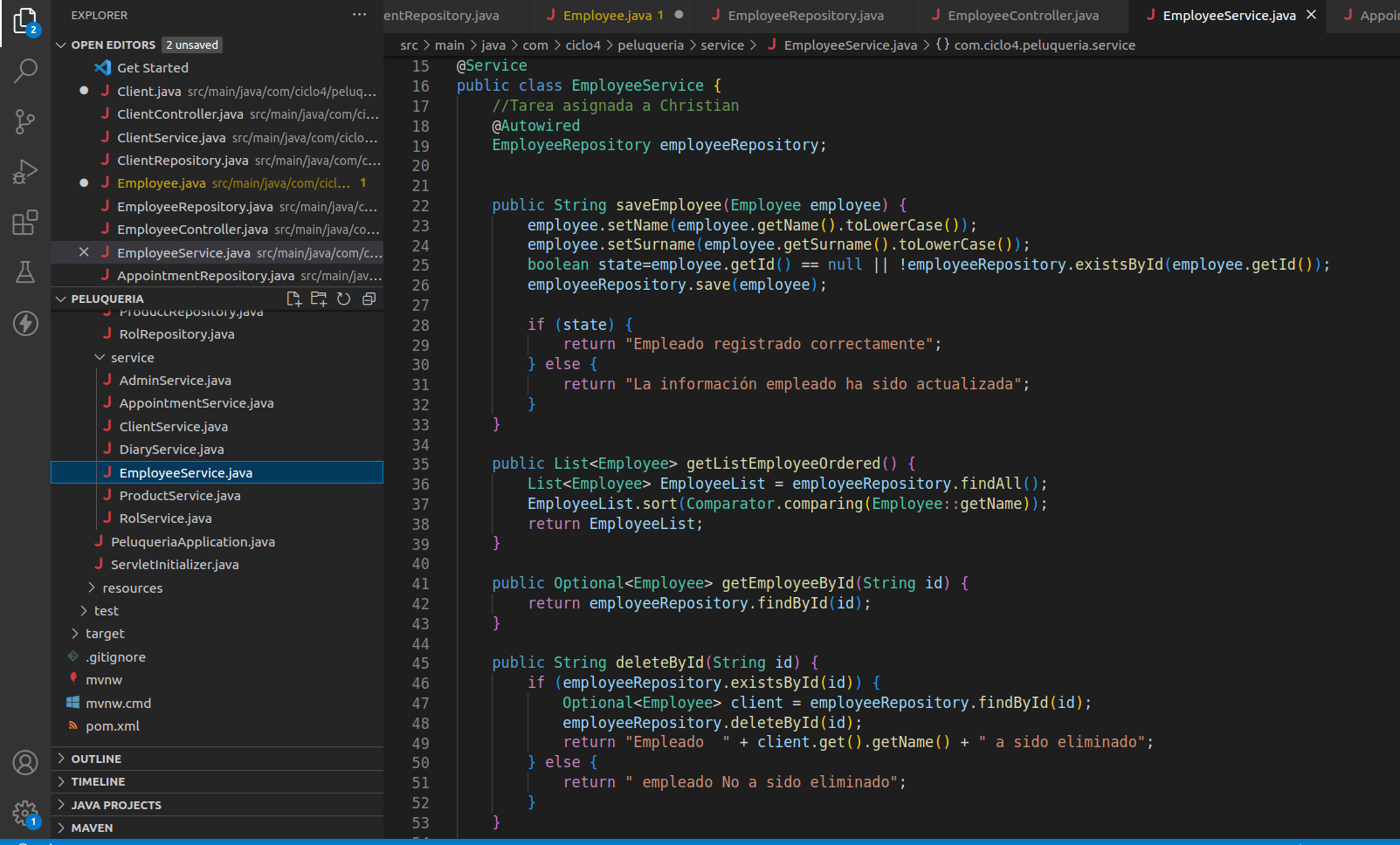


Creamos el repositorio (métodos CRUD) para el Employee:

Crear el controlador Employee:

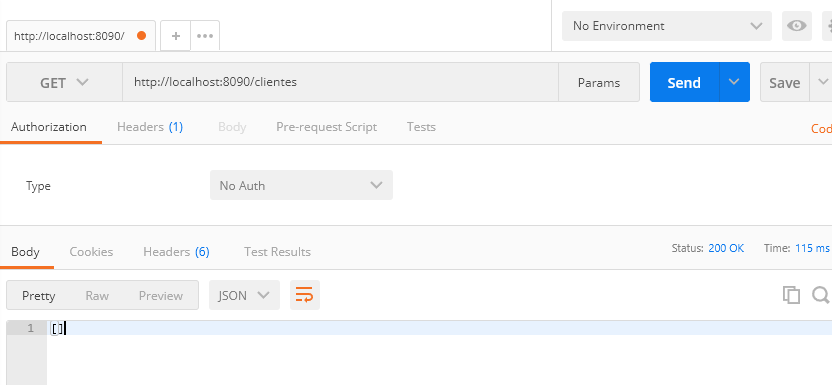


Creacion de servico Employee:

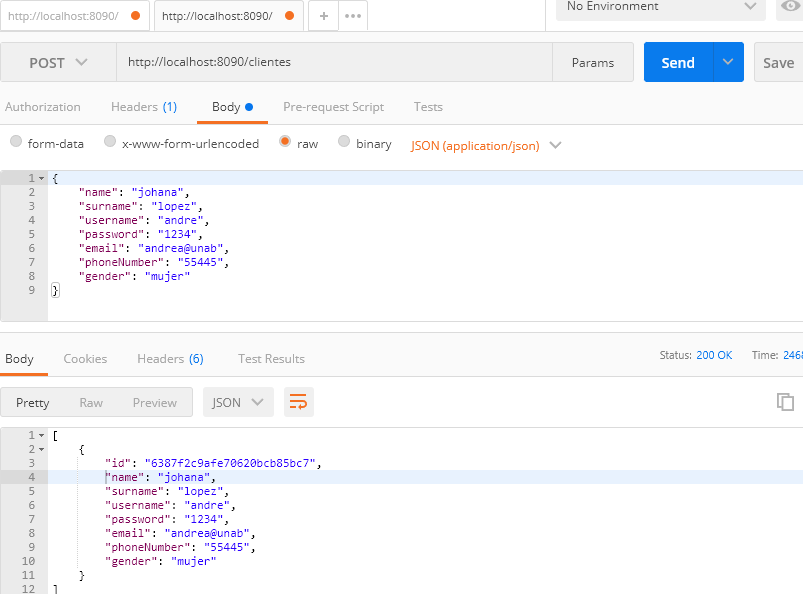


|  |
| --- |
| **Evidencias de los “endpoint” con el consumo de recursos del API REST** |

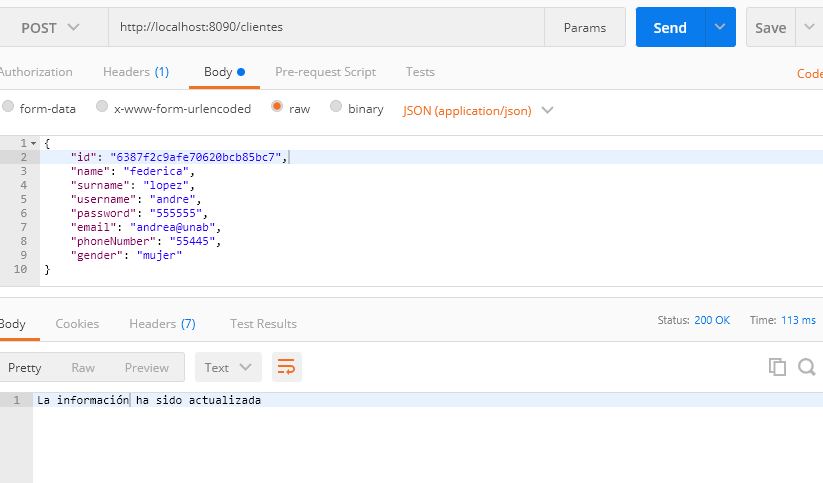
Método GET funcionando



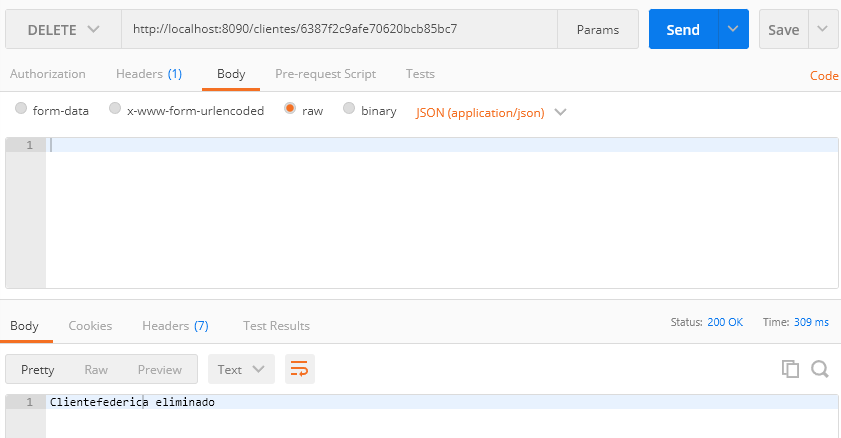
Método POST funcionando



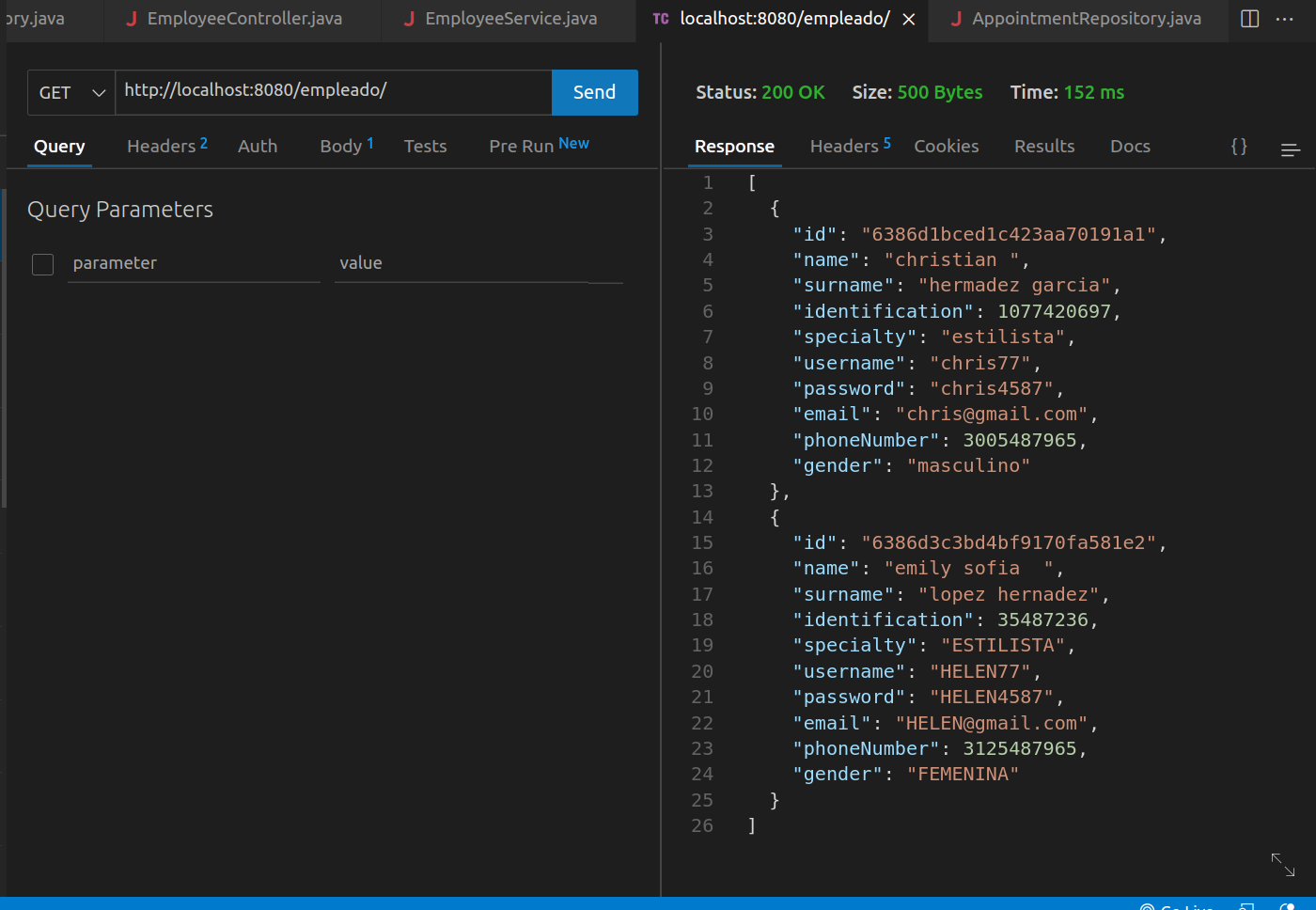
Metodo POST para actualizar



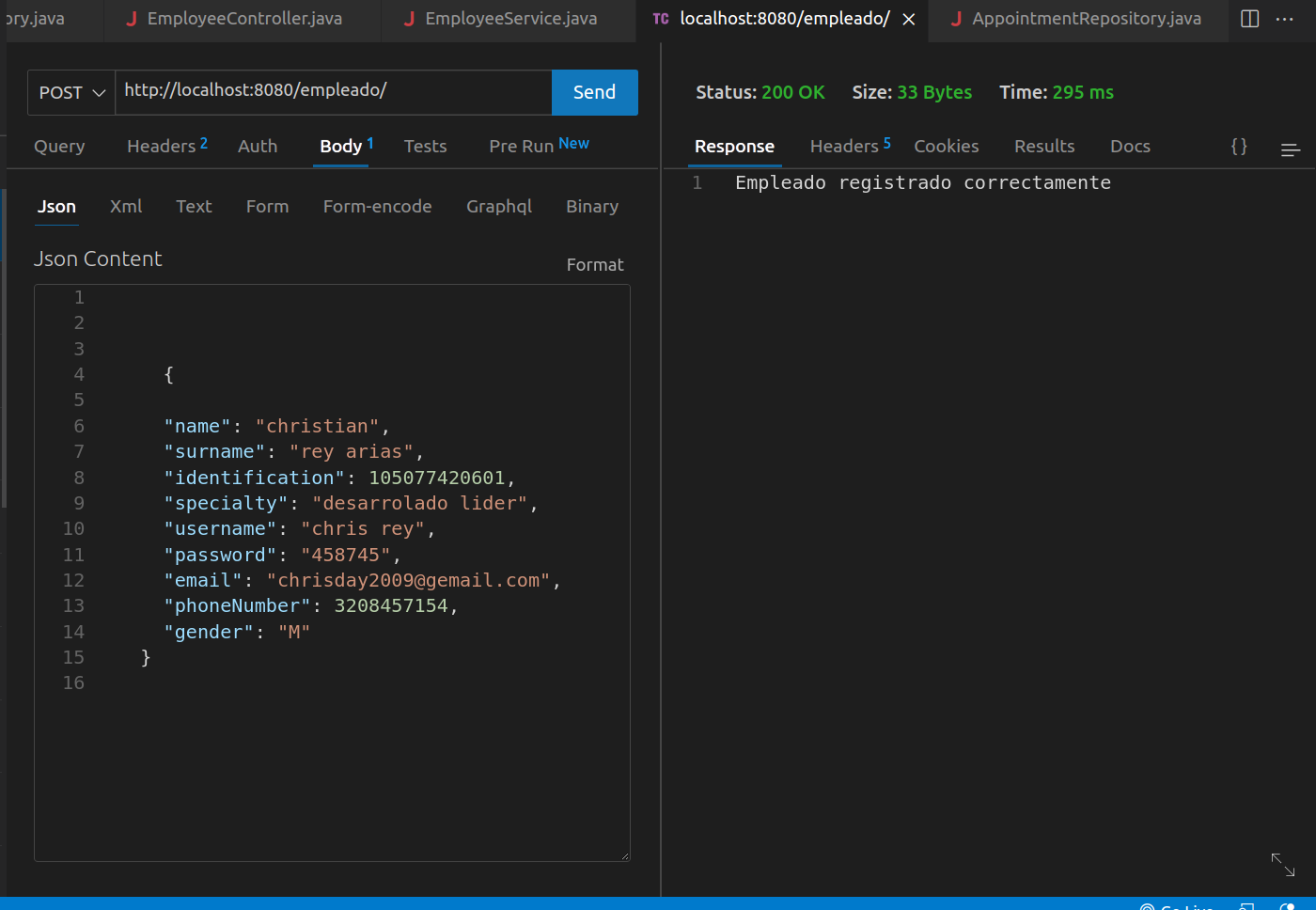
Método DELETE funcionando



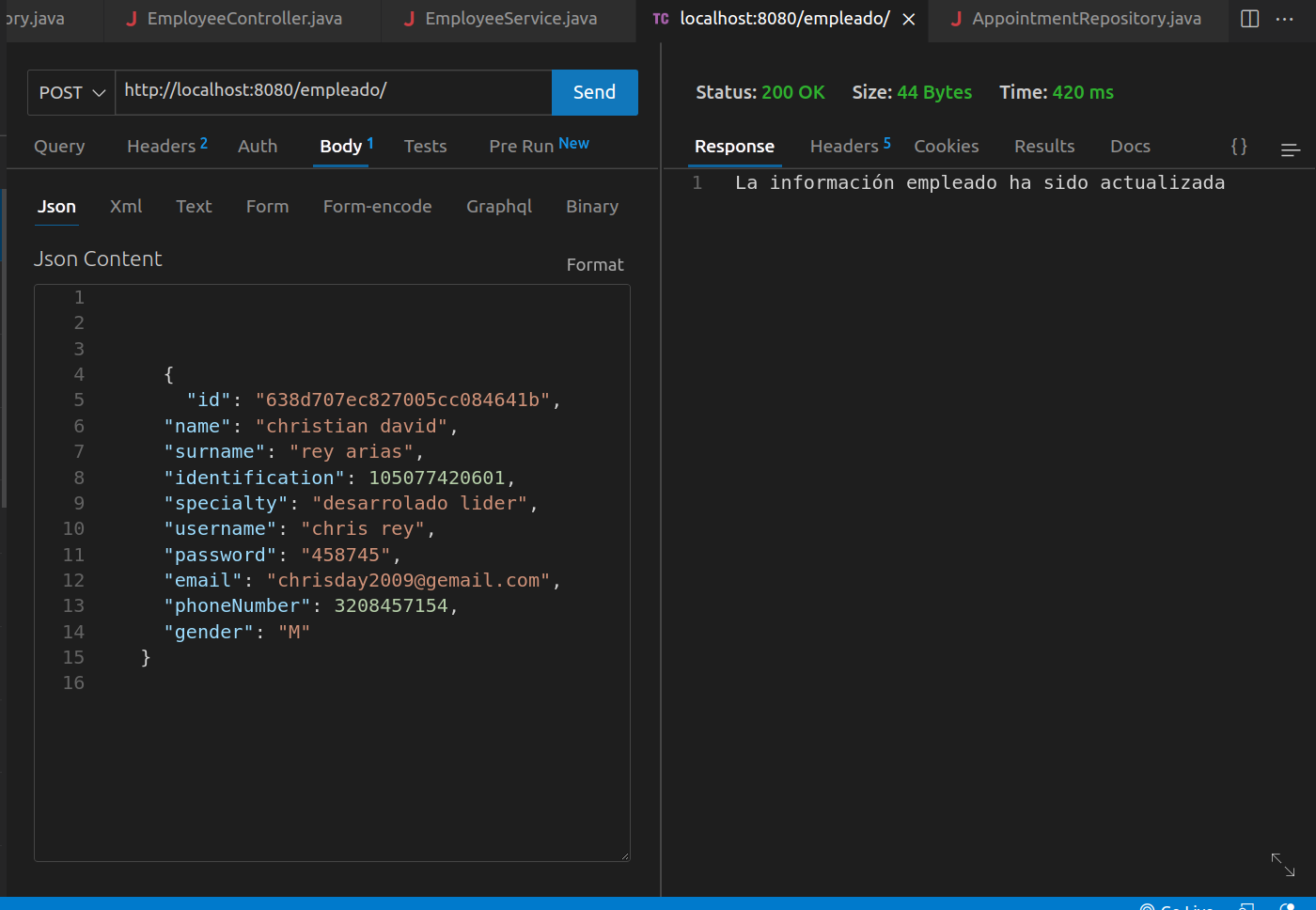
Método GET funcionando Empleado



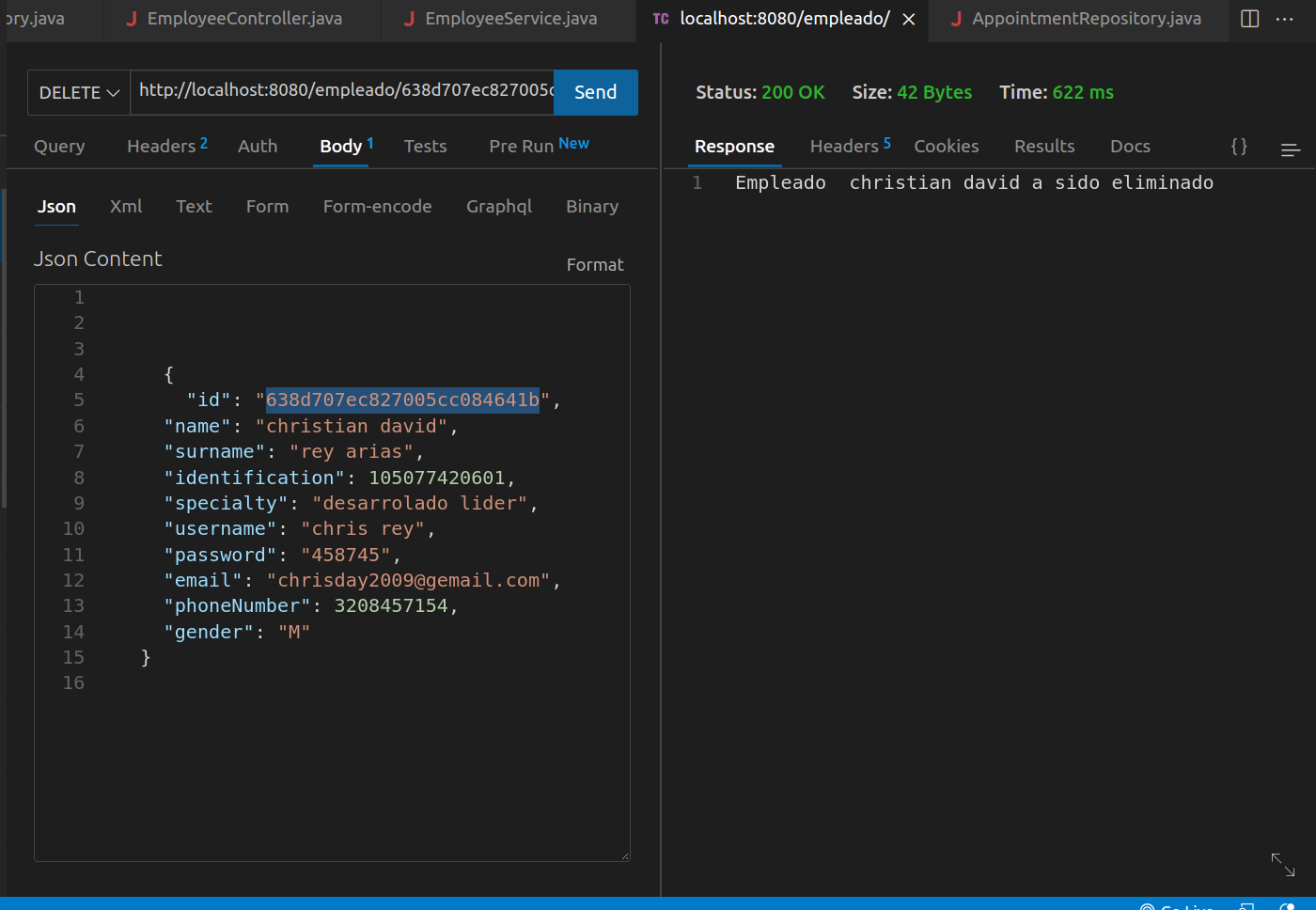
Método POST funcionando Empleado



Metodo POST para actualizar empleado



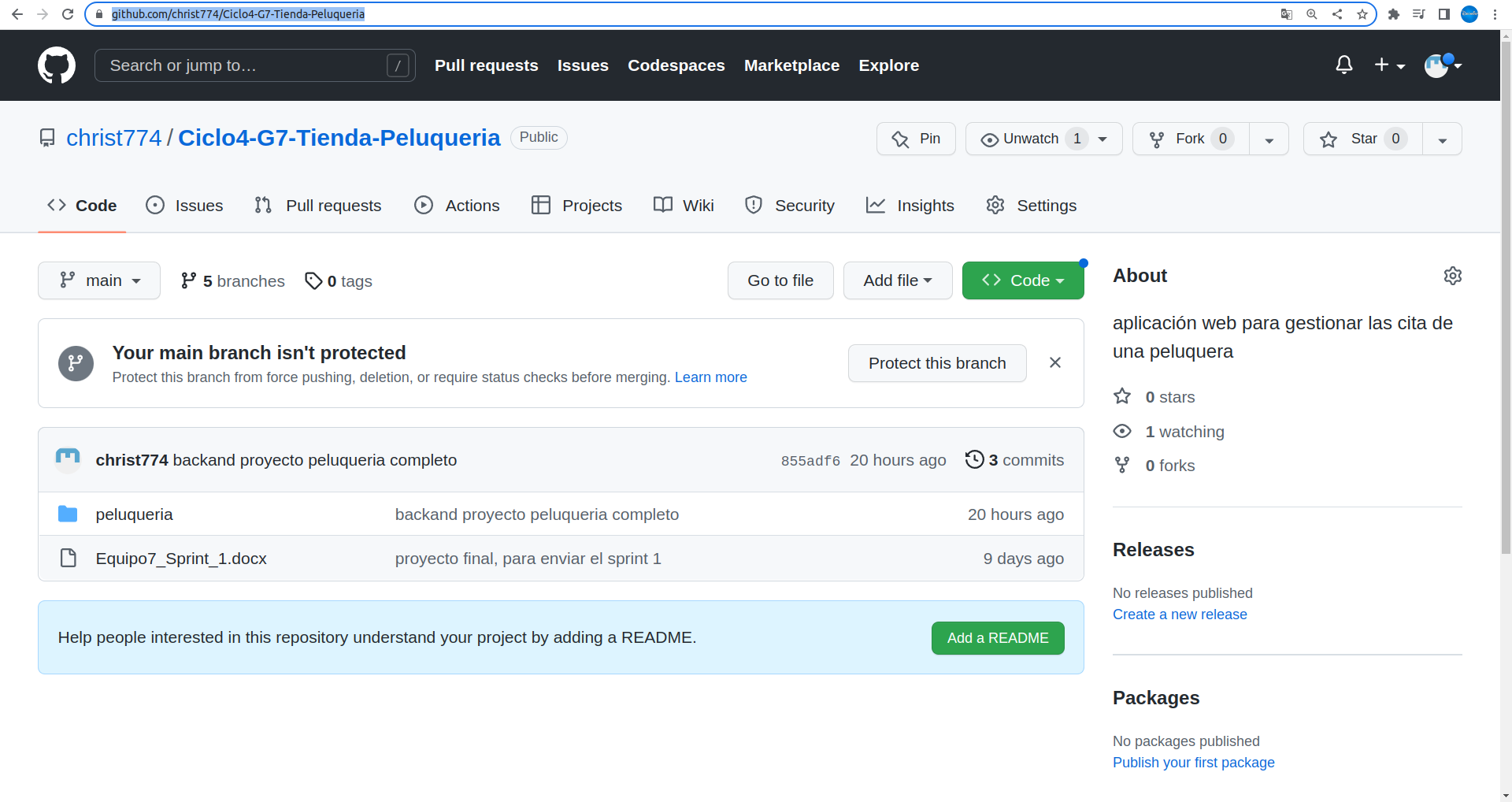
Método DELETE funcionando empleado

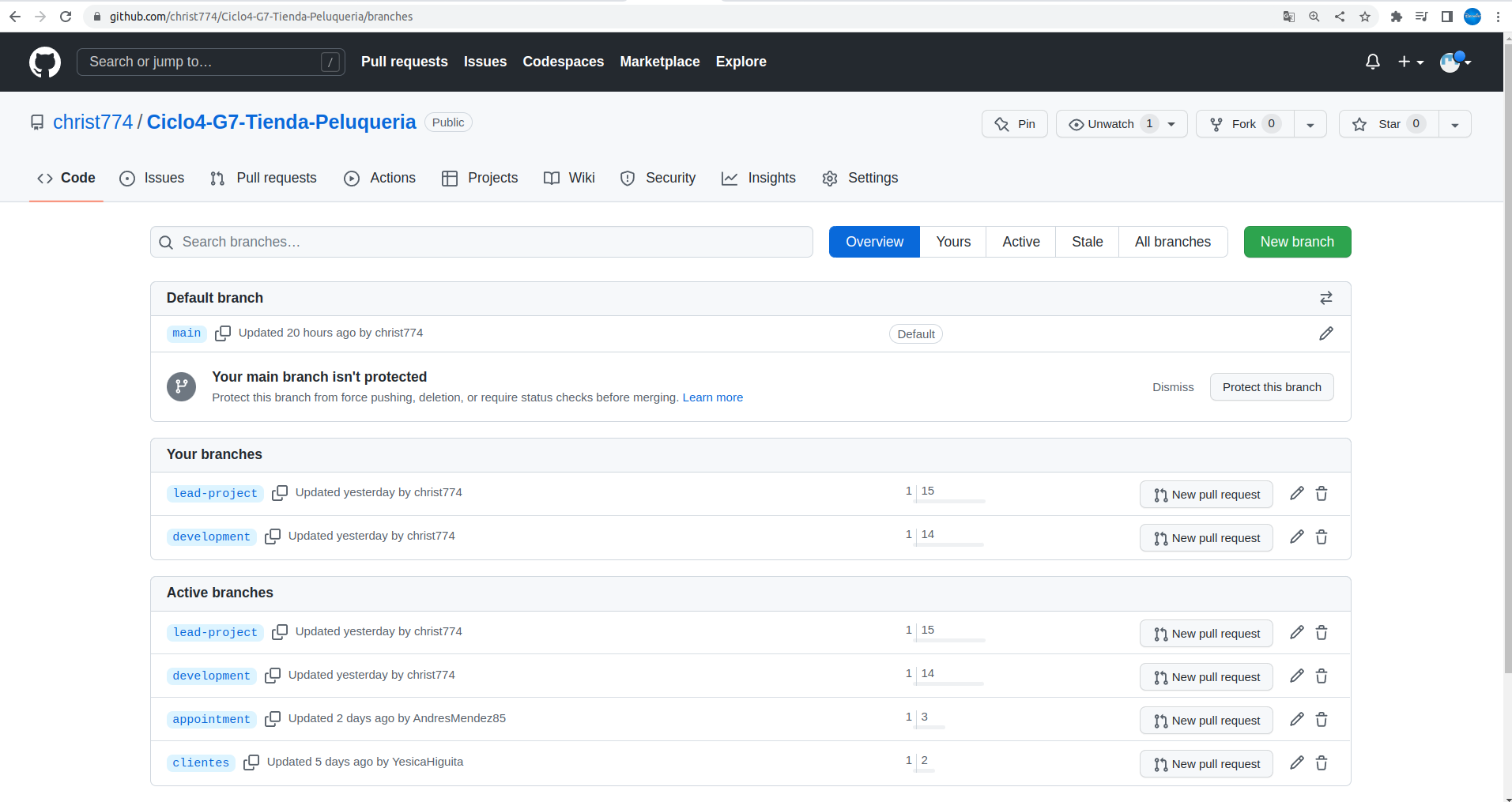


|  |
| --- |
| **Evidencia GitLab o GitHub** |

Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versión)

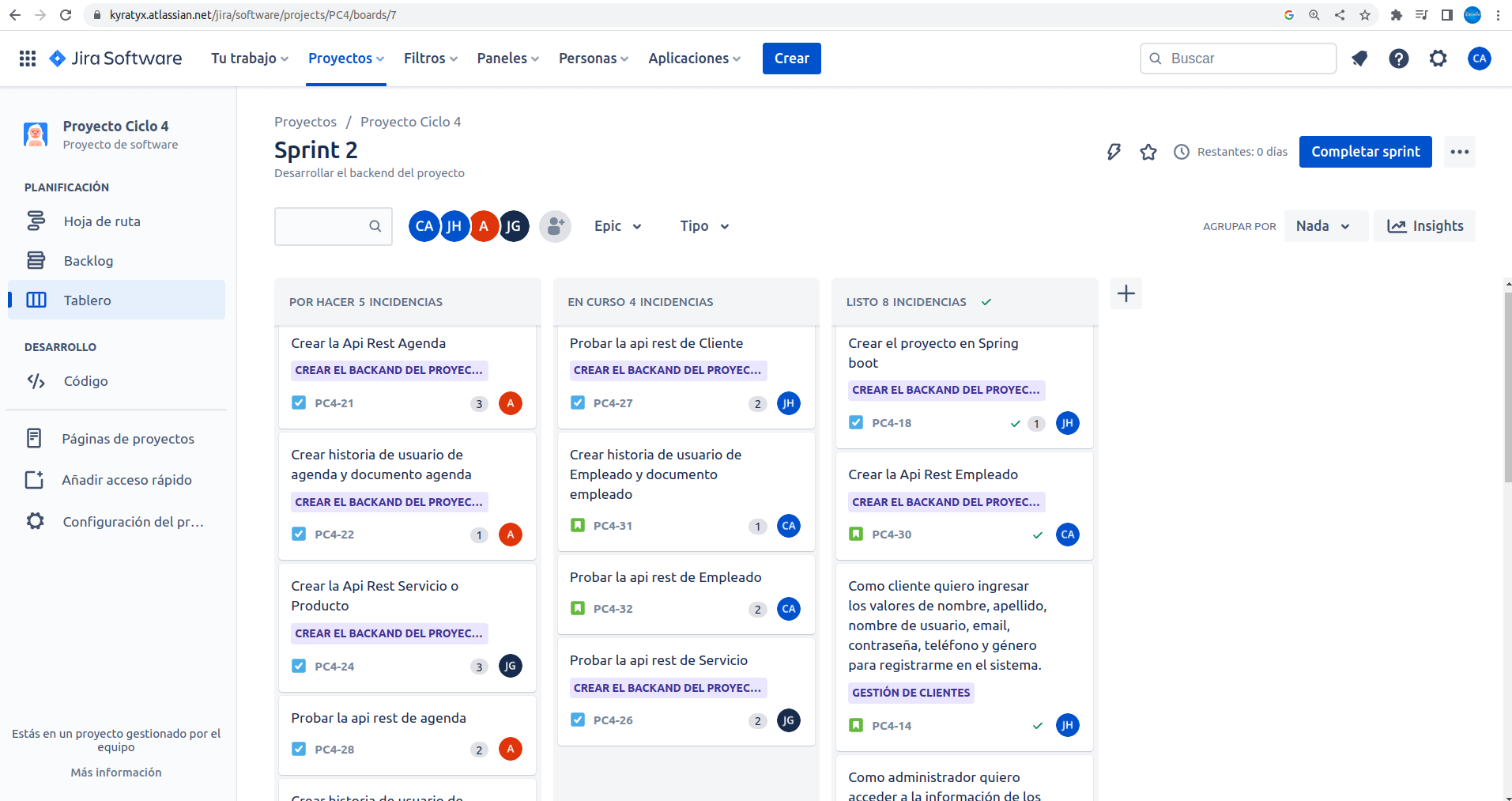
<https://github.com/christ774/Ciclo4-G7-Tienda-Peluqueria>

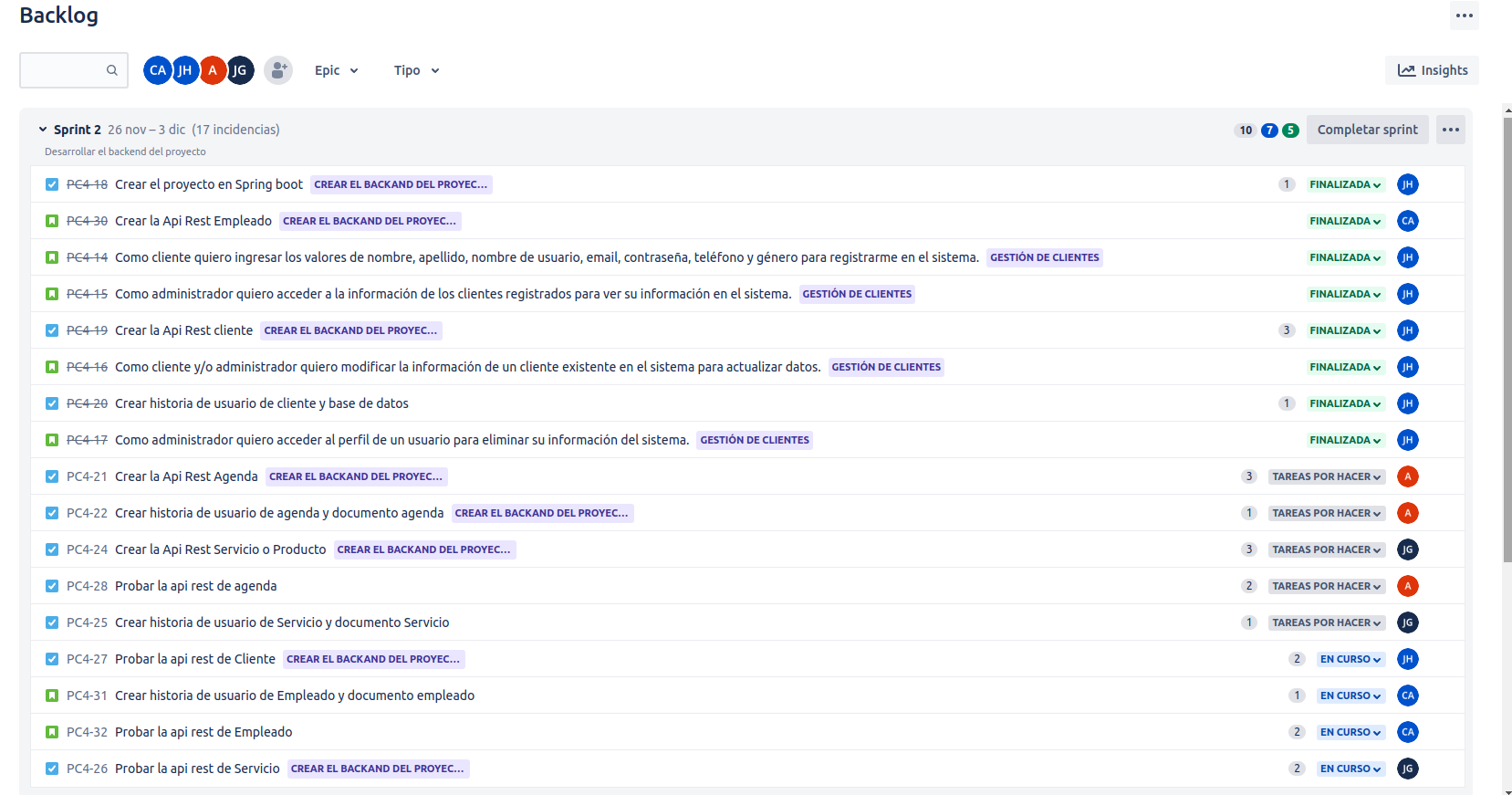


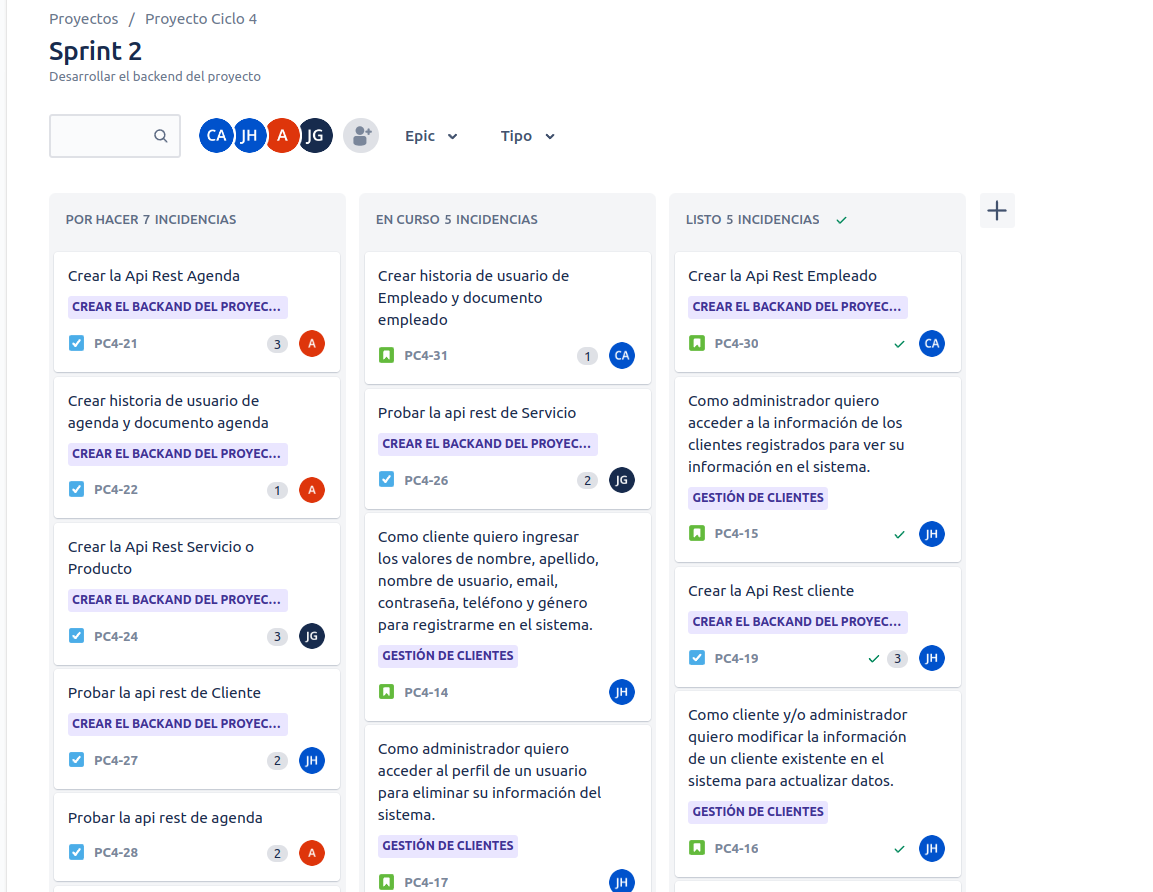


|  |
| --- |
| **Evidencia JIRA (Seguimiento del proyecto)** |

Como evidencia del seguimiento del proyecto con la metodología ágil SCRUM, utilizando el software JIRA, se debe presentar capturas de pantalla donde se visualice la ejecución de los Sprint con las historias de usuario relacionadas con el desarrollo del Backend.







|  |
| --- |
| **Evidencias de las Reuniones de Equipo** |

Como evidencia de las reuniones que efectúa el equipo del proyecto, presentar capturas de pantalla de las reuniones efectuadas y si lo consideran pertinente algunas actas de las reuniones.

