The code provided to students consists of the following packages and classes:

**• Package en.um.redes.nanoChat.client.application**

1. **NanoChat.java:** Main class of the client that has the sequence of calls to the controller. This class is already finished.
2. **NCController.java:** Implementation of the controller that will be in charge of coordinate communication with the directory and with the chat server depending on the phase in which we are or depending on what the user enters through the shell.

**• Package es.um.redes.nanoChat.client.shell**

1. **NCShell.java:** Shell implementation.
2. **NCCommands.java:** Support class to facilitate the implementation of the shell. Contains the definition of the types of commands and the parameters accepted by them.

**• Package en.um.redes.nanoChat.client.comm**

1. **NCConnector.java:** It proposes the functionality necessary to communicate the client with the chat server The exchanges of initiated messages must be programmed in it mainly by the chat client.

**• Package en.um.redes.nanoChat.directory.connector**

1. **DirectoryConnector.java:** Class implementation that will provide the necessary methods for communication with the Directory using UDP sockets. It will be used by both the client and the chat server.

**• Package en.um.redes.nanoChat.directory.server**

1. **Directory.java:** Implementation of the main class that processes call parameters to the directory server and is responsible for launching the process in background that will attend requests
2. **DirectoryThread.java:** Implementation of related functionality with the directory.

**• Package es.um.redes.nanoChat.messageFV**

1. **NCMessage.java:** Superclass that provides the abstraction of a message from the protocol between the client and the chat server. It contains useful methods for generate or interpret messages, some fully developed. This class is they will have to derive the different types of messages that are needed.
2. **NCRoomMessage.java:** Sample subclass that provides implementation specific of a specific type of message.

**• Package en.um.redes.nanoChat.server**

1. **nanoChatServer.java:** Main class of the chat server. It is responsible for creating the server socket, accept connections and generate the corresponding threads to attend incoming requests
2. **NCServerThread.java:** Class that implements the thread to be attended by each incoming connection from a client.
3. **NCServerManager.java:** Class used to manage the different rooms in the server as well as the different users that are connected to the server. An object of this class will be shared among all threads.

**• Package en.um.redes.nanoChat.server.roomManager**

1. **NCRoomManager.java:** Abstract class (superclass) to be used as the basis for the time to implement the specific logic of each chat. It will be inherited from it by implementing the necessary methods and attributes.
2. **NCRoomDescription.java:** Class that represents the description of the current state of a room, such as your name, the list of members and the date of the last message sent. It also contains a method to transform this information into a printable format per screen.