



Apéndice C. Envío de Bytes a través de la IR Tower

Fragmento de la clase Model.java:

```
public class Model extends Observable implements ActionCode {

    private boolean error = false;
    private String Command = "";
    private Tower tower;
    private static Model model = new Model();
    private Queue queue;

    /**
     * Constructor de la clase. Crea una sola conexión con el transmisor
     * de infrarrojos al utilizar un patrón Singleton. Utiliza el patrón
     * Observer para avisar a todas las vistas de los comandos que se
     * envíen.
     */
    private Model() {
        tower = new Tower();
        tower.open("USB");
        queue = new Queue(this);
        queue.start();
    }

    /**
     * Método implementa el patrón Singleton.
     * @return Model Regresa la única conexión con la IR Tower.
     */
    public static Model createModel(){
        return model;
    }

    /**
     * Método que envía un paquete de bytes al RCX por medio de la IR
     * Tower.
     * @param robot int ID del robot.
     * @param action int Número de identificación de la acción que el
     * robot va a realizar.
     */
    private void sendAction(int robot, int action) {
        byte[] byteToSend = stringToByteArray(robot + " " + action);
        tower.send(byteToSend, byteToSend.length);
        notifyChanges();
    }
}
```



```
/**
 * Método que inserta un comando en la cola de envío de mensajes al
 * RCX.
 * @param index int Posición de la cola en la que se va a insertar el
 * comando.
 * @param newInstruction Integer Comando que se va a insertar.
 */
public void insertInstruction(int index, Integer newInstruction){
    queue.insertInstruction(index, newInstruction);
}

/**
 * Método que puede poner o quitar la pausa de la cola de envío de
 * mensajes.
 * @param wait boolean Valor que indica si la cola se va a pausar o a
 * continuar.
 */
public void setWait(boolean wait){
    queue.setWait(wait);
}

/**
 * Método que le indica a las vistas de que un comando ha sido
 * enviado.
 */
public void notifyChanges(){
    setChanged();
    notifyObservers();
}
```