```java
package prg_trabajo2_hundirlaflota;

import java.util.Arrays;
import java.util.Scanner;

public class PRG_Trabajo2_HundirLaFlota {

    //=======================================
    // FUNCIÓN MAIN & MENÚ PRINCIPAL
    //=======================================
    public static void main(String[] args) {}

    // Muestra el menú y devuelve la opción elegida por el usuario (1, 2, 3 o 4)
    public static int menu_modo_juego() {}

    //=======================================
    // FUNCIÓNES DE JUGAR PARTIDA
    //=======================================

    // Juega una nueva partida en un tablero de tamaño filas x columnas,
    // con los barcos indicados y el nº máximo de max_disparos
    public static void juega_partida(int filas, int columnas, int lanchas, int buques, int acorazados, int portaaviones, int max_disparos) {}

    // Pide los param de una partida personalizada y los devuelve en un vector
    public static int[] pide_parametros_partida_personalizada() {}

    //=======================================
    // FUNCIONES DE TABLERO
    //=======================================

    // Crea y devuelve un tablero vacío (sin barcos) de tamaño filas x columnas
    // Contendrá '-' en todas sus posiciones
    public static char[][] crear_tablero(int filas, int columnas) {}

    // Muestra por pantalla el tablero completo (lo visible y lo oculto)
    public static void muestra_tablero_completo(char[][] tablero) {}

    // Muestra por pantalla el tablero visible al usuario
    public static void muestra_tablero_visible(char[][] tablero) {}

    //=======================================
    // FUNCIONES DE BARCOS
    //=======================================

    // Comprueba si queda algún barco en el tablero
    public static boolean quedan_barcos(char[][] tablero) {}

    // Inserta todos los barcos indicados en posiciones aleatorias del tablero
    public static void insertar_barcos(char[][] tablero, int lanchas, int buques, int acorazados, int portaaviones) {}

    // Inserta una lancha en una posición aleatoria del tablero
    public static void inserta_lancha(char[][] tablero) {}

    // Inserta un buque en una posición aleatoria del tablero (si cabe)
    public static void inserta_buque(char[][] tablero) {}

    // Inserta un acorazado en una posición aleatoria del tablero (si cabe)
    public static void inserta_acorazado(char[][] tablero) {}

    // Inserta un portaaviones en una posición aleatoria del tablero (si cabe)
    public static void inserta_portaaviones(char[][] tablero) {}

    // Comprueba si es posible insertar una lancha en la coordenada (f,c) del tablero
    public static boolean comprueba_inserta_lancha(char[][] tablero, int f, int c) {}

    // Comprueba si es posible insertar un buque en la coordenada (f,c) del tablero
    public static boolean comprueba_inserta_buque(char[][] tablero, int f, int c) {}

    // Comprueba si es posible insertar un acorazado en la coordenada (f,c) del tablero
    public static boolean comprueba_inserta_acorazado(char[][] tablero, int f, int c) {}

    // Comprueba si es posible insertar un portaaviones en la coordenada (f,c) del tablero
    public static boolean comprueba_inserta_portaaviones(char[][] tablero, int f, int c) {}

    //=======================================
    // FUNCIONES DE DISPARO
    //=======================================

    // Pide coordenadas de disparo (repite hasta que sean válidas)
    // y las devuelve en un vector de tamaño 2.
    public static int[] pide_coordenadas_disparo(char[][] tablero) {}

    // Realiza un disparo en las coordenadas indicadas
    public static void realiza_disparo(char[][] tablero, int f, int c) {}

    // Transforma la coordenada de la fila en int (A es 0, B es 1, C es 2, etc.)
    // Devuelve nº negativo si la cadena está mal formateada
    public static int fila_string_to_int(String s) {}

    //=======================================
    // FUNCIONES ÚTILES ADICIONALES
    //=======================================

    // Devuelve un int int_aleatorio entre min y max
    public static int int_aleatorio(int min, int max) {}

    // Devuelve una coordenada aleatoria del tablero (en un vector tamaño 2)
    public static int[] coordenada_aleatoria(char[][] tablero) {}

    // Comprueba si existe la coordenada en el tablero
    public static boolean existe_coordenada(char[][] tablero, int f, int c) {}

    // Muestra el texto al usuario, le pide por teclado un valor int y lo devuelve.
    // Se repite indefinidamente hasta que el valor introducido esté entre min y max
    public static int pide_int_forzoso_entre_min_y_max(String texto, int min, int max) {

}
```