```
#!/bin/bash
clear
#
#Comprobar si no se ha introducido ningun argumento
#
if [ $# -eq 0 ]; then
      echo No has introducido ningun argumento
      exit 1
fi
#
#Comprobar si se ha introducido más de un argumento
#
if [ $# -gt 1 ]; then
      echo Has introducido mas de un argumento
      exit 1
fi
#
#Comprobar si el unico argumento se llama conf.cfg
#
if [ "$1" != "conf.cfg" ]; then
      echo El argumento debe ser \"conf.cfg\"
      exit 1
fi

#
#Comprobaciones del fichero que contiene las estadísticas
#Inicio de la función testEstadisticas
#
testEstadisticas(){

      RUTA=(`pwd`)
      #
      #Coger la ultima linea del fichero conf.cfg, separar por dos puntos
      # y coger la segunda columna
      #
      ESTADISTICASPATH=(`tail -1 conf.cfg|cut -d ":" -f 2`)
      #
      #Comprobar que en la segunda columna hay algo, de no ser así saldrá
      #
      if [ ${#ESTADISTICASPATH} -eq 0 ]; then
            echo "No hay RUTA ni fichero correspondiente a estadisticas"
            exit 1
      fi
      #
      #Del resultado anterior coger hasta el ultimo / y usarlo como ruta
      # y coger desde el final hasta el primer / que se encuentre para usarlo
      # como nombre de fichero
      #
      UBIEST=${ESTADISTICASPATH%/*}
      FICEST=${ESTADISTICASPATH##*[/]}
      #
      #Comprobar que la ruta existe, en caso de que no exista utilizar
      # actual como ruta del fichero
      #
      if !(test -d $UBIEST); then
            UBIEST=$RUTA
      fi
      #
      #Comprobar que existe un fichero llamado como se especifica en el
      # directorio que se ha especificado, en caso de que no exista crea
      # un fichero con ese nombre en la ruta
      #
      #En caso de que exista se comprobará si se tienen permisos de lectura
      # si no se tienen se sale del script
```

```
	#
	#En caso de que exista y se tengan permisos de lectura se comprobará
	# que se tienen permiso para escribir en él
	#
	if !(test -f "$UBIEST/$FICEST"); then
		touch "$UBIEST/$FICEST"
		chmod 755 "$UBIEST/$FICEST"
		rm conf.cfg
		touch conf.cfg
		chmod 744 conf.cfg
		echo DICCIONARIO: $UBIDIC/$FICDIC>>conf.cfg
		echo ESTADISTICAS: $UBIEST/$FICEST>>conf.cfg
		echo "Se ha creado el fichero de estadisticas "\"$FICEST\"" en el
directorio "\"$UBIEST\"
	else
		if !(test -r $ESTADISTICASPATH); then
			echo No tienes permiso para leer el fichero \"$FICEST\"
			exit 1
		fi
		if !(test -w $ESTADISTICASPATH); then
			echo No tienes permiso para escribir en el fichero \"$FICEST\"
			exit 1
		fi
	fi
}
#
#Fin de la funcion testEstadisticas
#
#Comprobar el fichero que contendr?á las palabras del diccionario
#Inicio de la funcion testDiccionario
#
testDiccionario(){

	#
	#Coger la primera linea del fichero conf.cfg, separar por dos puntos
	# y quedarnos con la segunda columna
	#
	DICCIONARIOPATH=(`head -1 conf.cfg|cut -d ":" -f 2`)
	#
	#Comprobar que en la segunda columna hay algo, de no se así se saldrá
	#
	if [ ${#DICCIONARIOPATH} -eq 0 ]; then
		echo "No hay RUTA ni fichero correspondiente a diccionario"
	    exit 1
	fi
	#
	#Del resultado anterior cortar hasta el ultimo / y guardarlo como ruta
	# y cortar desde el final hasta el primer / que se encuentre como
	# el fichero a usar como nombre de fichero
	#
	UBIDIC=${DICCIONARIOPATH%/*}
	FICDIC=${DICCIONARIOPATH##*[/]}
	#
	#Comprobar que existe el fichero especificado en la ruta especificada
	#
	if !(test -f $DICCIONARIOPATH); then
	echo El fichero \"diccionario.txt\" no esta en el directorio o no existe,
creelo antes de ejecutar el script
		exit 1
	fi
	#
	#Comprobar que tenemos permisos de lectura sobre el fichero
	#
	if !(test -r $DICCIONARIOPATH); then
```

```
            echo No tienes permiso para leer el fichero \"$FICDIC\"
      exit 1
    fi
}
#
#Fin de la funcion testDiccionario
#
#Realizar todas las comprobaciones oportunas
#Inicio de la funcion tests
#
tests(){
      #
      #Comprobar que el fichero conf.cfg existe y no esta vacio
      #
      if !(test -s "conf.cfg"); then
            echo El fichero \"conf.cfg\" no esta en el directorio, no existe o
esta vacio creelo antes de ejecutar el script
            exit 1
      fi
      #
      #Comprobar que tenemos permiso para leer el fichero conf.cfg
      #
      if !(test -r "conf.cfg"); then
            echo No tienes permisos para leer \"conf.cfg\"
            exit 1
      fi
      #
      #Comprobar el fichero que contiene el diccionario
      #
      testDiccionario
      #
      #Comprobar el fichero que contiene las estadisticas
      #
      testEstadisticas
}
#
#Fin de la funcion tests
#
#Creacion de un menu con todas las opciones
#Inicio de la funcion menu
#
menu(){
      clear
      #
      #Realizar las comprobaciones de todos los ficheros
      #
      tests

      echo "         PALABRATOR"
      echo "========================="
      echo "J) JUGAR"
      echo "E) ESTADISTICAS"
      echo "C) CONFIGURACION"
      echo "A) AYUDA"
      echo "G) GRUPO"
      echo "S) SALIR"
      echo ""
      printf "Juego Palabrator, introduzca opcion >> "
      read
      #
      #Recoges de teclado la opcion y la comparas buscando una coincidencia
      # en caso de no encontrar ninguna vuelve al menu
      #
      case "$REPLY" in
```

```
            j|J)
                  clear
                  game
                  askContinue;;
            e|E)
                  statistics
                  askContinue;;
            c|C)
                  config
                  askContinue;;
            a|A)
                  clear
                        help
                  askContinue;;
            g|G)
                  clear
            echo "--------------------------------------------"
            echo "                   GRUPO"
            echo "--------------------------------------------"
            echo ""
                  echo "   Juan Carlos Martin Garcia   70882826T"
                  echo "   Mercedes Parra Sanchez      70938751N"
            echo ""
                  askContinue;;
            s|S)
                  echo Has salido del menu
                  exit 0;;
            *)
                  echo Has introducido mal la opcion, prueba de nuevo
                  askContinue;;
      esac
}
#
#End of menu function and begining of game function
#
game(){

      #
      #Variables que contendrán todas las consonantes y vocales que se
      # usarán
      #
      CONSONANTES=bcdfghjklmnpqrstvwxyz
      VOCALES=aeiou
      #
      #Las variables Letra[0] y Letra[1] son vocales, escogidas aleatoriamente
      # de la variable VOCALES, logrando que no sean la misma vocal
      #
      LETRA[0]=${VOCALES:($RANDOM % 5):1}
      LETRA[1]=${VOCALES:($RANDOM % 5):1}

      while test ${LETRA[0]} = ${LETRA[1]}
      do
            LETRA[1]=${VOCALES:($RANDOM % 5):1}
      done
      #
      #A partir de Letra[2] a Letra[5] son consonantes elegidas de forma
      # aleatoria de la variable CONSONANTES, logrando que cada una sea
diferente
      # al resto
      #
      LETRA[2]=${CONSONANTES:($RANDOM % 21):1}
      for i in {3..5}
      do
            ENCONTRADO=1
```

```
            until test $ENCONTRADO -eq 0
            do
                  LETRA[$i]=${CONSONANTES:($RANDOM % 21):1}
                  ENCONTRADO=0
                  for ((j=2; j<$i ;j++))
                  do
                  if [ ${LETRA[$i]} = ${LETRA[$j]} ]; then
                        ENCONTRADO=1
                  fi
            done
      done
      done
      #
      #Variables que se usaran para escribir en el fichero que contiene
      # las estadisticas
      #
      FALLOS=0
      INTENTO=0
      PUNTUACION=0
      PALABRASVALIDAS=""
      PALABRASFALLIDAS=""
      CADENA=""
      DAY=(`date +%d-%m-%Y`)
      TIEMPOINICIO=(`perl -e "print time"`)
      #
      #Variable que almacena el PID del proceso
      #
      PID=(`ps |grep -i "palabr"|cut -d " " -f 2`)
      #
      #Llamada a la funcion que nos permitirá jugar
      #
      bucleJuego
}

bucleJuego(){
      #
      #Comprobar si ya se han producido los 3 fallos, si ya
      # se ha llegado a los 10 aciertos
      # en ese caso se escribirá en el fichero estadisticas
      #
      if [ $FALLOS -eq 3 ]; then
            echo "El juego ha terminado"
        echo "Se ha excedido el numero de fallos"
            writeResults

      elif [ $INTENTO -eq 10 ]; then
            echo "El juego ha terminado"
        echo "Se ha excedido el numero de intentos"
            writeResults
      fi
      #
      #Mostramos el menu oportuno
      #
      clear
      echo "--------------------------------------------"
      echo "                  JUGAR"
      echo "--------------------------------------------"
      echo ""

      echo "VOCALES: ${LETRA[0]} , ${LETRA[1]}"
      echo "CONSONANTES: ${LETRA[2]} , ${LETRA[3]} , ${LETRA[4]} , ${LETRA[5]}"
      echo ""
      printf "Nueva Palabra >> "
      read NUEVAPALABRA
```

```
      #
      #Comprobamos que la palabra no sea una palbra vacia, que sea un punto
      # o que sea un caracter fuera del rango a-z ó A-Z
      #En cualquier caso no se contara ni como fallo ni como intento y se
volvera
      # a mostrar todo, a excepcion del punto, que hará que se escriba en ls
      # estadisticas
      #
    if [ ${#NUEVAPALABRA} -eq 0 ]; then
            showGame
      elif [ $NUEVAPALABRA = "." ]; then
            echo Saliendo del juego...
            if [ $INTENTO -eq 0 ]; then
                  askContinue
            else
                  writeResults
            fi
      elif [ $NUEVAPALABRA = "$( echo $NUEVAPALABRA | egrep '[^a-zA-Z]' )" ];
then
            echo La palabra tiene un caracter invalido
            showGame
      fi

      #
      #Si no es el primer intento se comprobar?á la palabra escrita con las
anteriores
      # en caso de que esté reptida no se contabiliza ni el fallo ni el intento
      # mostrando en ese caso el men?u con todas las letras
      #
    if [ $INTENTO -ne 0 ]; then

            for ((i=0;i<${#CADENA[@]};i++))
            do
                  if [ ${CADENA[$i]} = $NUEVAPALABRA ]; then
                        echo "La palabra "${CADENA[$i]}" ya se ha repetido,
pruebe de nuevo"
                        showGame
                  fi
            done
      fi
      CADENA[$INTENTO]="$NUEVAPALABRA"

      let INTENTO++
      #
      #Cambia las mayusculas por minusculas de la palabra escrita
      #
      NUEVAPALABRA=(`echo $NUEVAPALABRA | tr '[A-Z]' '[a-z]'`)
      #
      #Comprueba letra a letra de la palabra escrita si es una letra permitida
en esa ejecucion
      # en caso de que no sea una letra valida ENCONTRADO será 0, se volverá a
mostrar el mu
      # y se contabilizara como fallo
      #
      for ((i=0;i<${#NUEVAPALABRA};i++))
      do
            ENCONTRADO=0
            for j in {0..5}
            do
                  if [ ${NUEVAPALABRA:i:1} = ${LETRA[j]} ]; then
                        ENCONTRADO=1
                  fi
            done
```

```
            if [ $ENCONTRADO -eq 0 ]; then
                  echo "La letra \"${NUEVAPALABRA:i:1}\" no se encuentra entre
las letras del juego "
                  if [ ${#PALABRASFALLIDAS} -eq 0 ]; then
                        PALABRASFALLIDAS=$NUEVAPALABRA
                  else
                        PALABRASFALLIDAS="$PALABRASFALLIDAS,$NUEVAPALABRA"
                  fi
                  let FALLOS++
                  showGame
            fi
      done
      #
      #Comprobamos permisos del diccionario y buscamos la palabra en el
ignorando las mayusculas
      # y encontrando la palabra al completo, es decir, que no este contenida en
otra
      #
      #Una vez comprobada la almacenamos en el array correspondiente, VALIDAS o
FALLIDAS
      #
      testDiccionario
      flag=(`grep -iw "$NUEVAPALABRA" $DICCIONARIOPATH`)
      if [ $? -ne 0 ]; then
            echo "La palabra $NUEVAPALABRA no se encuentra en el diccionario"
            if [ ${#PALABRASFALLIDAS} -eq 0 ]; then
                  PALABRASFALLIDAS=$NUEVAPALABRA
            else
                  PALABRASFALLIDAS="$PALABRASFALLIDAS,$NUEVAPALABRA"
            fi
                  let FALLOS++
                  showGame
      else
            echo "Enhorabuena!!"
            if [ ${#PALABRASVALIDAS} -eq 0 ]; then
                  PALABRASVALIDAS=$NUEVAPALABRA
            else
                  PALABRASVALIDAS="$PALABRASVALIDAS,$NUEVAPALABRA"
            fi
            PUNTUACION=$(($PUNTUACION+${#NUEVAPALABRA}))
            showGame
      fi
}

showGame(){

      echo ""
      echo "Letras: ${LETRA[0]} ${LETRA[1]} ${LETRA[2]} ${LETRA[3]} ${LETRA[4]}
${LETRA[5]}"
      echo "Nº de intento: $INTENTO"
      echo "Listado de palabras validas: $PALABRASVALIDAS "
      echo "Listado palabras fallidas: $PALABRASFALLIDAS"
      echo "Puntuacion total: $PUNTUACION"
      echo ""
      echo ""
      echo \"Pulsa una tecla para continuar\"
      read
      bucleJuego
}

writeResults(){

      #
      #Comprobamos cuantos segundos han pasado desde el inicio de la partida y
```

```
escribimos
      # en el fichero estadisticas
      #
      TIEMPOFIN=(`perl -e "print time"`)
      TIEMPO=$(($TIEMPOFIN - $TIEMPOINICIO))
      HORA=(`date +%H:%M`)

      testEstadisticas

      echo "$PID:$DAY:$HORA:$PUNTUACION:$TIEMPO:$PALABRASVALIDAS:
$PALABRASFALLIDAS" >> "$UBIEST/$FICEST"

      askContinue
}
#
#End of function game and begining of statistics function
#
statistics(){
      clear
      #
      #estadisticas.txt tests
      #
      testEstadisticas
      #
      #That actualizes our variables UBIEST and FICEST
      #
      #CONTENIDO aloja la primera linea de las estadisticas, lo usamos para
comprobar
      # si la longitud de CONTENIDO es igual a 0 y saber en ese caso que el
fichero
      # de estadisticas esta vacio
      #
      CONTENIDO=(`cat $ESTADISTICASPATH | head -1`)
      #
      #Make sure the first line of estadisticas.txt is not empty
      #
      if [ ${#CONTENIDO} -eq 0 ]; then
            LINEA="/"
            LINEAPUNTUACIONMAX="/"
            LINEAPUNTUACIONMIN="/"
            LINEATIEMPOMAX="/"
            LINEATIEMPOMIN="/"
            LINEAMEGAPALABRA="/"
      else
            LINEA=0
            #
            #Leer del fichero
            #
            while IFS=: read partida fecha HORA minuto PUNTUACION TIEMPO VALIDAS
fallidas
            do

                  LINEA=$(($LINEA+1))
                  #
                  #Si el numero de linea es uno hay que establecer los valores
iniciales
                  #
                  if [ $LINEA -eq 1 ]; then
                        PUNTUACIONMAX=$PUNTUACION
                        PUNTUACIONMIN=$PUNTUACION
                        TIEMPOMAX=$TIEMPO
                        TIEMPOMIN=$TIEMPO
                        MEGAPALABRALONGITUD=0
                  fi
```

```
                    #
                    #Si el tiempo actual es mayor o igual que el tiempo maximo
                    # establece el tiempo maximo como el tiempo actual y actualiza
                    # la linea del tiempo maximo
                    #
                    if [ $TIEMPO -ge $TIEMPOMAX ]; then
                        TIEMPOMAX=$TIEMPO
                        LINEATIEMPOMAX=$LINEA
                    fi
                    #
                    #Lo mismo que arriba, pero para el minimo
                    #
                    if [ $TIEMPO -le $TIEMPOMIN ]; then
                    TIEMPOMIN=$TIEMPO
                          LINEATIEMPOMIN=$LINEA
                    fi
                    #
                    #Lo mismo que arriba, pero con la puntuacion maxima
                    #
                    if [ $PUNTUACION -ge $PUNTUACIONMAX ]; then
                        PUNTUACIONMAX=$PUNTUACION
                        LINEAPUNTUACIONMAX=$LINEA
                    fi
                    #
                    #Lo mismo que arriba, pero con la puntuacion minima
                    #
                    if [ $PUNTUACION -le $PUNTUACIONMIN ]; then
                        PUNTUACIONMIN=$PUNTUACION
                        LINEAPUNTUACIONMIN=$LINEA
                    fi
                    #
                    #Establecer PALABRA como palabra a usar la primera palabra de
las
                    # validas y el resto de las validas en RESTOPALABRA
                    #
                    PALABRA=${VALIDAS%%,*}
                    RESTOPALABRA=${VALIDAS#*,}
                    #
                    #Comprobar una a una las palabras que estan dentro de la
cadena validas
                    # en caso de que la longitud de una de las palabras sea mayor
a la de la
                    # megapalabra se actualiza la linea en la que se aloja la
megapalabra a
                    # la linea actual, repetira el proceso hasta que PALABRA sea
una cadena vacia
                    #
                    while (test -n $PALABRA); do
                        if [ ${#PALABRA} -gt $MEGAPALABRALONGITUD ]; then
                          MEGAPALABRALONGITUD=${#PALABRA}
                          LINEAMEGAPALABRA=$LINEA
                        fi

                          PALABRA=${RESTOPALABRA%%,*}
                          RESTOPALABRA=${RESTOPALABRA#*,}
                          if [ $PALABRA=$RESTOPALABRA ]; then
                                break
                          fi
                    done

                done < "$ESTADISTICASPATH"
                #
                #Seleccion de las lineas que contienen la puntuacion max, min,
tiempo max, min
```

```
            # y la linea que contiene la megapalabra
            #
            LINEAPUNTUACIONMAX=(`sed -n $(($LINEAPUNTUACIONMAX))p "$UBIEST/
$FICEST"`)
            LINEAPUNTUACIONMIN=(`sed -n $(($LINEAPUNTUACIONMIN))p "$UBIEST/
$FICEST"`)
            LINEATIEMPOMAX=(`sed -n $(($LINEATIEMPOMAX))p "$UBIEST/$FICEST"`)
            LINEATIEMPOMIN=(`sed -n $(($LINEATIEMPOMIN))p "$UBIEST/$FICEST"`)
            LINEAMEGAPALABRA=(`sed -n $(($LINEAMEGAPALABRA))p "$UBIEST/
$FICEST"`)

      fi

      #
      #Muestra las estadisticas
      #
    clear
    echo "-----------------------------------------"
    echo "               ESTADISTICAS"
    echo "-----------------------------------------"
    echo "Los datos de la partida vienen en el siguiente formato "
    echo "Partida : Fecha : Hora : Puntuacion : Tiempo : Palabras Validas :
Palabras Fallidas"
    echo ""
      echo "TOTALES"
      echo "-----------------------"
      echo "Numero total de partidas jugadas: "$LINEA
      echo ""
      echo "JUGADAS ESPECIALES"
      echo "-----------------------"
      echo "Datos de la jugada con mayor puntuacion: "
    echo "        $LINEAPUNTUACIONMAX"
      echo ""
      echo "Datos de la jugada mas corta: "
    echo "        $LINEATIEMPOMIN"
      echo ""
      echo "Datos de la jugada con peor puntuacion: "
    echo "        $LINEAPUNTUACIONMIN"
      echo ""
      echo "Datos de la jugada mas larga: "
    echo "        $LINEATIEMPOMAX"
      echo ""
      echo "MEGAPALABRA"
      echo "-----------------------"
      echo "Datos de la jugada con la palabra valida mas larga: "
    echo "        $LINEAMEGAPALABRA"
      echo ""

}
#
#End of function statistics and begiging of config function
#
config(){

    clear
    echo "-----------------------------------------"
    echo "               CONFIGURACION"
    echo "-----------------------------------------"
    echo ""
    echo "Que desea cambiar?"
    echo "1. El fichero diccionario a usar"
    echo "2. El fichero estadisticas a usar"
    echo "q. Salir"
    echo ""
```

```
    printf "Introduzca una opcion >> "
    read
    echo ""
    #
    #Comprobamos que todos los ficheros estan bien
    #
    tests
      #
      #Comprobamos primero que realmente quiere cambiar el fichero X a usar
      # no se modificara si no se introduce s/S
      #
    case "$REPLY" in
        1)
            clear
            echo "La RUTA del actual fichero es: "$DICCIONARIOPATH
            echo Quieres utilizar otro fichero diccionario? \(s/n\)
            read
            #
            #Se pide la ruta de un nuevo diccionario a usar
            #
            case $REPLY in
                  s|S)
                        echo Introduce la ruta junto al nombre del nuevo
diccionario que vas a usar
                        read NEWDICCIONARIOPATH
                        #
                        #Se comprueba que existe y que es diferente a la
anterior ruta
                        #
                        if !(test -f $NEWDICCIONARIOPATH); then
                              echo $NEWDICCIONARIOPATH" no existe el fichero"
                        askContinueConfig
                        else
                              if [ "$UBIDIC/$FICDIC" != $NEWDICCIONARIOPATH ];
then
                                    case "$NEWDICCIONARIOPATH" in
                                          */*)
                                                #
                                                #Se comprueban los permisos de
lectura
                                                #
                                          if !(test -r "$NEWDICCIONARIOPATH");
then
                                                      echo No tienes permisos
para leer el fichero
                                                askContinueConfig
                                                fi
                                                #
                                                #Se elimina y se crea el fichero
conf.cfg actualizado
                                                #
                                                rm conf.cfg
                                                touch conf.cfg
                                                chmod 744 conf.cfg
                                                echo "DICCIONARIO:
"$NEWDICCIONARIOPATH>>conf.cfg
                                    echo "ESTADISTICAS:
"$ESTADISTICASPATH>>conf.cfg
                                                ;;
                                          *)
                                                #
                                                #Lo mismo que arriba pero
teniendo en cuenta que es
                                                # en el directorio actual
```

```
                                                #
                                                if !(test -r
"$NEWDICCIONARIOPATH"); then
                                                      echo No tienes permisos
para leer el fichero
                                                askContinueConfig
                                                fi
                                                rm conf.cfg
                                                touch conf.cfg
                                                chmod 744 conf.cfg
                                                echo "DICCIONARIO:
"$RUTA"/"$NEWDICCIONARIOPATH>>conf.cfg
                                          echo "ESTADISTICAS:
"$ESTADISTICASPATH>>conf.cfg
                                                ;;
                                          esac
                                          echo Se ha cambiado el fichero diccionario a
usar
                                    else
                              echo El fichero con el diccionario se encuentra en la
misma direccion que antes
                        fi
                              fi
                        askContinueConfig
                        ;;
                        *)
                              echo No se produciran cambios en el fichero
                              askContinueConfig
                              ;;
                  esac

                  askContinueConfig
                  ;;
            2)
                  clear
                        echo "La RUTA del actual fichero es: "$ESTADISTICASPATH
                        echo Quieres utilizar otro fichero estadisticas? \(s/n\)
                        read
                        #
                        #Se pide la ruta con el fichero estadisticas que se va a usar
                        #
                        case $REPLY in
                              s|S)
                                    clear
                                    echo Introduce la ruta junto al nombre de las
nuevas estadisticas que vas a usar
                              read NEWESTADISTICASPATH
                                    #
                                    #Se comprueba que realmente es un fichero y que se
tienen permisos sufientes
                                    # tanto para escribir como para leer de el
                                    #
                                    if !(test -f $NEWESTADISTICASPATH); then
                                          echo $NEWESTADISTICASPATH" no existe el
fichero o la ruta que lleva a el"
                                          askContinueConfig
                                    else
                                          if [ "$UBIEST/$FICEST" !=
$NEWESTADISTICASPATH ]; then
                                          case "$NEWESTADISTICASPATH" in
                                                      */*)
                                                            if !(test -r
"$NEWESTADISTICASPATH"); then
                                                                  echo No tienes
```

```
permisos para leer el fichero
                                                      askContinueConfig
                                                fi
                                                if !(test -w
"$NEWESTADISTICASPATH"); then
                                                      echo No tienes
permisos para escribir en el fichero
                                                      askContinueConfig
                                                fi
                                                #
                                                #Se elimina y se crea
conf.cfg actualizado
                                                #
                                                rm conf.cfg
                                                touch conf.cfg
                                                chmod 744 conf.cfg
                                                echo "DICCIONARIO:
"$DICCIONARIOPATH>>conf.cfg
                                                echo "ESTADISTICAS:
"$NEWESTADISTICASPATH>>conf.cfg
                                                ;;
                                          *)
                                          if !(test -r
"$NEWESTADISTICASPATH"); then
                                                      echo No tienes
permisos para leer el fichero
                                          askContinueConfig
                                                fi
                                                if !(test -w
"$NEWESTADISTICASPATH"); then
                                                      echo No tienes
permisos para escribir en el fichero
                                                      askContinueConfig
                                                fi
                                                #
                                                #Igual que arriba pero
teniendo en cuenta que la ruta
                                                # hace referencia al
directorio en el cual se trabaja
                                                #
                                                rm conf.cfg
                                                touch conf.cfg
                                                chmod 744 conf.cfg
                                                echo "DICCIONARIO:
"$DICCIONARIOPATH>>conf.cfg
                                                echo "ESTADISTICAS:
"$RUTA"/"$NEWESTADISTICASPATH>>conf.cfg
                                     ;;
                                           esac
                                           echo Se ha cambiado el fichero
estadisticas a usar
                        else
                              echo El fichero con las estadisticas se encuentra
en la misma direccion que antes
                        fi
                              fi
                              ;;
                        *)
                              echo No se produciran cambios en el fichero
                              askContinueConfig
                              ;;
                  esac

            askContinueConfig
```

```
            ;;
        q|Q)
            echo Saliendo de la configuracion...
            askContinue
            ;;
            *)
            echo La opcion no es valida
            askContinueConfig
                  ;;
    esac
}
#
#End of config function and begining of help function
#
help(){
    clear
      #
      #Generamos un fichero con toda la ayuda
      #
    echo "-------------------------------------------" >>help.txt
    echo "                   AYUDA" >>help.txt
    echo "-------------------------------------------" >>help.txt
    echo "Para el correcto funcionamiento del programa se le debe pasar como
argumento un" >>help.txt
    echo "fichero conf.cfg donde se especifica la ruta del diccionario y las
estadisticas" >>help.txt
    echo "Para continuar con el juego debe haber un fichero diccionario en la
ruta" >>help.txt
    echo "especificada en conf.cfg" >>help.txt
    echo "Si no existe el fichero estadisticas el programa crea uno nuevo en la
ruta" >>help.txt
    echo "descrita en conf.cfg" >>help.txt
    echo "Al abrir el juego aparecen en pantalla 6 opciones" >>help.txt
    echo "J) JUGAR: " >>help.txt
    echo "==========" >>help.txt
    echo "En esta opcion del juego se desarrolla una partida de Palabrator."
>>help.txt
    echo "El juego muestra 4 consonates y 2 vocales, el jugador debera ingresar
una" >>help.txt
    echo "palabra, dicha palabra debe constar de las letras mostradas."
>>help.txt
    echo "En el caso que esa palabra se encuentre en el diccionario asociado al
juego" >>help.txt
    echo "la palabra se contabiliza como valida, cada palabra ingresada es un
intento y " >>help.txt
    echo "la puntuacion del intento es el numero de letras de la palabra."
>>help.txt
    echo "Puede ocurrir que:" >>help.txt
    echo "  * No haya ningun diccionario asociado, en este caso el juego
termina" >>help.txt
    echo "  * La palabra contega una letra que no ha mostrado el juego entonces
la" >>help.txt
    echo "    palabra se contabiliza como fallida" >>help.txt
    echo "  * La palabra contenga un caracter no valido, es decir, un caracter
diferente" >>help.txt
    echo "    del alfabeto, en este caso el juego vuelve a mostrar las letras"
>>help.txt
    echo "  * La palabra no se encuentra en el diccionario entonces es una
palabra fallida" >>help.txt
    echo "Al final de cada palabra ingresada se muestra un listado con las
letras mostradas" >>help.txt
    echo "el numero de intentos, el listado de palabras validas, el listado de
palabras" >>help.txt
    echo "fallidas y la puntuacion total" >>help.txt
```

```
    echo "El juego termina cuando el usuario ingresa un punto, el numero de
intentos es" >>help.txt
    echo "mayor que 10 o el numero de fallos es mayor que 3" >>help.txt
    echo "E) ESTADISTICAS: " >>help.txt
    echo "=================" >>help.txt
    echo "Se muestran los datos de:" >>help.txt
    echo "  * La jugada con mayor puntuacion" >>help.txt
    echo "  * La jugada mas corta, con duracion menor" >>help.txt
    echo "  * La jugada con peor puntuacion" >>help.txt
    echo "  * La jugada mas larga, con duracion mayor" >>help.txt
    echo "  * La jugada con la palabra valida mas larga" >>help.txt
    echo "El formato de la jugada es: " >>help.txt
    echo "Partida : Fecha : Hora : Puntuacion : Tiempo : Palabras Validas :
Palabras Fallidas" >>help.txt
    echo "Si el fichero estadisticas no contiene nada muestra una barra (/)"
>>help.txt
    echo "C) CONFIGURACION: " >>help.txt
    echo "==================" >>help.txt
    echo "El programa muestra un menu de que desea cambiar" >>help.txt
    echo "    1. El fichero diccionario a usar" >>help.txt
    echo "    2. El fichero estadisticas a usar" >>help.txt
    echo "    q. Salir" >>help.txt
    echo "Si se introduce la opcion 1, el programa muestra al usuario la ruta
donde se" >>help.txt
    echo "se encuentra el diccionario y le indica si la quiere cambiar, en caso
afirmativo" >>help.txt
    echo "el usuario debe introducir una ruta valida" >>help.txt
    echo "Si se introduce la opcion 2, el programa muestra al usuario la ruta
donde se" >>help.txt
    echo "se encuentran las estadisticas y le indica si la quiere cambiar, en
caso afirmativo" >>help.txt
    echo "el usuario debe introducir una ruta valida" >>help.txt
    echo "para salir se debe introducir la letra q" >>help.txt
    echo "A) AYUDA: " >>help.txt
    echo "==========" >>help.txt
    echo "muestra la ayuda para utilizar el programa correctamente" >>help.txt
    echo "G) GRUPO: " >>help.txt
    echo "==========" >>help.txt
    echo "Indica el nombre y el DNI de los componentes del grupo" >>help.txt
    echo "S) SALIR: " >>help.txt
    echo "==========" >>help.txt
    echo "Para salir del programa" >>help.txt
    echo "" >>help.txt
    echo "" >>help.txt
      #
      #Mostramos el fichero poco a poco y tras una lectura del mismo lo
eliminamos
      #
    more help.txt
    rm help.txt
}

askContinueConfig(){
    echo ""
    echo \"Pulsa una tecla para continuar\"
    read
    config
}
#
#End of askContinueConfig function and begining of askContinue function to ask
to exit
#
askContinue(){
      echo ""
```

```
      echo \"Pulsa una tecla para continuar\"
      read
      menu
}
#
#End of askContinue function
#
#
#El script empieza con la llamada a la función menu
#
menu
```