

Simulador de Carreras

Generated by Doxygen 1.8.11

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	_Apostador Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	ben	5
3.1.2.2	din_rest	5
3.1.2.3	nombre	6
3.1.2.4	pid	6
3.1.2.5	total	6
3.2	_Apuesta Struct Reference	6
3.2.1	Detailed Description	7
3.2.2	Member Data Documentation	7
3.2.2.1	apos	7
3.2.2.2	c	7
3.2.2.3	cab_id	7
3.2.2.4	cantidad	7
3.2.2.5	ventanilla	7
3.3	_Caballo Struct Reference	7

3.3.1	Detailed Description	8
3.3.2	Member Data Documentation	8
3.3.2.1	cotizacion	8
3.3.2.2	dinero_apostado	8
3.3.2.3	id	8
3.3.2.4	last_tir	8
3.3.2.5	pid	8
3.3.2.6	posicion	8
3.4	_ventanilla Struct Reference	9
3.4.1	Detailed Description	9
3.4.2	Member Data Documentation	9
3.4.2.1	apostadores	9
3.4.2.2	caballos	9
3.4.2.3	msgqid	9
3.4.2.4	ventanilla	10
3.5	msgapues Struct Reference	10
3.5.1	Detailed Description	10
3.5.2	Member Data Documentation	10
3.5.2.1	caballo	10
3.5.2.2	cantidad	10
3.5.2.3	mtype	10
3.5.2.4	nombre	11
3.6	msgtir Struct Reference	11
3.6.1	Detailed Description	11
3.6.2	Member Data Documentation	11
3.6.2.1	mtype	11
3.6.2.2	tirada	11

4 File Documentation	13
4.1 apostador.c File Reference	13
4.1.1 Detailed Description	14
4.1.2 Function Documentation	14
4.1.2.1 apos_cmp_ben(const void *v1, const void *v2)	14
4.1.2.2 apos_destroy(Apostador *a)	15
4.1.2.3 apos_get_ben(Apostador *a, int id)	15
4.1.2.4 apos_get_name(Apostador *a)	15
4.1.2.5 apos_get_pid(Apostador *a)	15
4.1.2.6 apos_get_total(Apostador *a)	16
4.1.2.7 apos_incr_din_rest(Apostador *a, double delta)	16
4.1.2.8 apos_init(Apostador *a, char *name, double din_init)	16
4.1.2.9 apos_new()	17
4.1.2.10 apos_refresh_total(Apostador *a)	17
4.1.2.11 apos_set_ben(Apostador *a, double beneficio, int id)	17
4.1.2.12 apos_set_pid(Apostador *a, pid_t pid)	17
4.2 apostador.h File Reference	18
4.2.1 Detailed Description	19
4.2.2 Macro Definition Documentation	19
4.2.2.1 APOS_ERROR	19
4.2.2.2 MAX_APOS_NAME	20
4.2.2.3 MAX_CAB	20
4.2.3 Typedef Documentation	20
4.2.3.1 Apostador	20
4.2.4 Function Documentation	20
4.2.4.1 apos_cmp_ben(const void *a1, const void *a2)	20
4.2.4.2 apos_destroy(Apostador *a)	20
4.2.4.3 apos_get_ben(Apostador *a, int id)	21
4.2.4.4 apos_get_name(Apostador *a)	21
4.2.4.5 apos_get_pid(Apostador *a)	21

4.2.4.6	<code>apos_get_total(Apostador *a)</code>	21
4.2.4.7	<code>apos_incr_din_rest(Apostador *a, double din)</code>	22
4.2.4.8	<code>apos_init(Apostador *a, char *nombre, double din)</code>	22
4.2.4.9	<code>apos_new()</code>	22
4.2.4.10	<code>apos_refresh_total(Apostador *a)</code>	23
4.2.4.11	<code>apos_set_ben(Apostador *a, double ben, int id)</code>	23
4.2.4.12	<code>apos_set_pid(Apostador *a, pid_t id)</code>	23
4.3	<code>apuesta.c</code> File Reference	24
4.3.1	Detailed Description	25
4.3.2	Function Documentation	25
4.3.2.1	<code>_apuesta_print(FILE *fp, Apuesta *a, double old_cot)</code>	25
4.3.2.2	<code>apuesta_destroy(Apuesta *a)</code>	25
4.3.2.3	<code>apuesta_execute(Apuesta *a, char *path)</code>	25
4.3.2.4	<code>apuesta_init(Apuesta *a, Apostador *apos, Caballo *c, int cab_id, unsigned short ventanilla, double cantidad)</code>	27
4.3.2.5	<code>apuesta_new()</code>	27
4.3.3	Variable Documentation	27
4.3.3.1	<code>apuesta_total</code>	27
4.4	<code>apuesta.h</code> File Reference	28
4.4.1	Detailed Description	28
4.4.2	Typedef Documentation	29
4.4.2.1	<code>Apuesta</code>	29
4.4.3	Function Documentation	29
4.4.3.1	<code>apuesta_destroy(Apuesta *a)</code>	29
4.4.3.2	<code>apuesta_execute(Apuesta *a, char *filepath)</code>	29
4.4.3.3	<code>apuesta_init(Apuesta *a, Apostador *apos, Caballo *c, int cab_id, unsigned short ventanilla, double cantidad)</code>	29
4.4.3.4	<code>apuesta_new()</code>	30
4.5	<code>caballo.c</code> File Reference	30
4.5.1	Detailed Description	31
4.5.2	Function Documentation	31

4.5.2.1	cab_destroy(Caballo *c)	31
4.5.2.2	cab_get_apostado(Caballo *c)	32
4.5.2.3	cab_get_cot(Caballo *c)	32
4.5.2.4	cab_get_id(Caballo *c)	32
4.5.2.5	cab_get_last_tir(Caballo *c)	32
4.5.2.6	cab_get_pid(Caballo *c)	33
4.5.2.7	cab_get_pos(Caballo *c)	33
4.5.2.8	cab_incr_apostado(Caballo *c, double delta)	33
4.5.2.9	cab_new()	34
4.5.2.10	cab_set_cot(Caballo *c, double cotizacion)	34
4.5.2.11	cab_set_id(Caballo *c, unsigned short id)	34
4.5.2.12	cab_set_last_tir(Caballo *c, unsigned short tir)	34
4.5.2.13	cab_set_pid(Caballo *c, pid_t pid)	36
4.5.2.14	cab_set_pos(Caballo *c, unsigned int posicion)	36
4.6	caballo.h File Reference	36
4.6.1	Detailed Description	38
4.6.2	Macro Definition Documentation	38
4.6.2.1	CAB_ERROR	38
4.6.3	Typedef Documentation	38
4.6.3.1	Caballo	38
4.6.4	Function Documentation	38
4.6.4.1	cab_destroy(Caballo *c)	38
4.6.4.2	cab_get_apostado(Caballo *c)	39
4.6.4.3	cab_get_cot(Caballo *c)	39
4.6.4.4	cab_get_id(Caballo *c)	39
4.6.4.5	cab_get_last_tir(Caballo *c)	40
4.6.4.6	cab_get_pid(Caballo *c)	40
4.6.4.7	cab_get_pos(Caballo *c)	40
4.6.4.8	cab_incr_apostado(Caballo *c, double incr)	40
4.6.4.9	cab_new()	41

4.6.4.10	<code>cab_set_cot(Caballo *c, double cot)</code>	41
4.6.4.11	<code>cab_set_id(Caballo *c, unsigned short id)</code>	41
4.6.4.12	<code>cab_set_last_tir(Caballo *c, unsigned short tir)</code>	42
4.6.4.13	<code>cab_set_pid(Caballo *c, pid_t pid)</code>	42
4.6.4.14	<code>cab_set_pos(Caballo *c, unsigned int pos)</code>	42
4.7	<code>mylib.c</code> File Reference	43
4.7.1	Detailed Description	43
4.7.2	Function Documentation	44
4.7.2.1	<code>aredigits(const char *string)</code>	44
4.7.2.2	<code>isfloat(const char *string)</code>	44
4.7.2.3	<code>randNum(float inf, float sup)</code>	44
4.7.2.4	<code>sigaddset_var(sigset_t *sigt, int sig,...)</code>	45
4.8	<code>mylib.h</code> File Reference	45
4.8.1	Detailed Description	45
4.8.2	Function Documentation	46
4.8.2.1	<code>aredigits(const char *string)</code>	46
4.8.2.2	<code>isfloat(const char *string)</code>	46
4.8.2.3	<code>randNum(float inf, float sup)</code>	46
4.8.2.4	<code>sigaddset_var(sigset_t *sigt, int sig,...)</code>	47
4.9	<code>rutina_apostador.c</code> File Reference	47
4.9.1	Detailed Description	48
4.9.2	Function Documentation	48
4.9.2.1	<code>_apos_handler(int sig)</code>	48
4.9.2.2	<code>proc_apostador(int id)</code>	48
4.9.3	Variable Documentation	49
4.9.3.1	<code>apostadores</code>	49
4.9.3.2	<code>running_apostador</code>	49
4.10	<code>rutina_apostador.h</code> File Reference	49
4.10.1	Detailed Description	49
4.10.2	Function Documentation	50

4.10.2.1	<code>proc_apostador(int id)</code>	50
4.11	<code>rutina_gestor.c</code> File Reference	50
4.11.1	Detailed Description	51
4.11.2	Function Documentation	51
4.11.2.1	<code>_gestor_handler(int sig)</code>	51
4.11.2.2	<code>_rutina_ventanilla(void *data)</code>	51
4.11.2.3	<code>proc_gestor()</code>	52
4.11.3	Variable Documentation	52
4.11.3.1	<code>end</code>	52
4.12	<code>rutina_gestor.h</code> File Reference	52
4.12.1	Detailed Description	52
4.12.2	Function Documentation	53
4.12.2.1	<code>proc_gestor()</code>	53
4.13	<code>rutina_monitor.c</code> File Reference	53
4.13.1	Detailed Description	54
4.13.2	Macro Definition Documentation	54
4.13.2.1	<code>MAX_APOS_PRINT</code>	54
4.13.3	Function Documentation	54
4.13.3.1	<code>_monitor_carrera(Caballo *c)</code>	54
4.13.3.2	<code>_monitor_fin()</code>	54
4.13.3.3	<code>_monitor_handler(int sig)</code>	55
4.13.3.4	<code>_monitor_post_carrera(Caballo *c, Apostador *a)</code>	55
4.13.3.5	<code>_monitor_pre_carrera(Caballo *c)</code>	55
4.13.3.6	<code>proc_monitor()</code>	55
4.13.4	Variable Documentation	55
4.13.4.1	<code>fin_carr</code>	55
4.13.4.2	<code>fin_pre_carr</code>	56
4.14	<code>rutina_monitor.h</code> File Reference	56
4.14.1	Detailed Description	56
4.14.2	Function Documentation	57

4.14.2.1	<code>proc_monitor()</code>	57
4.15	<code>rutina_tirada.c</code> File Reference	57
4.15.1	Detailed Description	58
4.15.2	Function Documentation	58
4.15.2.1	<code>_tirada_handler(int sig)</code>	58
4.15.2.2	<code>proc_tirada(int id, int *pipe)</code>	58
4.15.3	Variable Documentation	58
4.15.3.1	<code>running_tirada</code>	58
4.16	<code>rutina_tirada.h</code> File Reference	59
4.16.1	Detailed Description	59
4.16.2	Function Documentation	59
4.16.2.1	<code>proc_tirada(int id, int *pipe)</code>	59
4.17	<code>semaforos.c</code> File Reference	60
4.17.1	Detailed Description	61
4.17.2	Function Documentation	61
4.17.2.1	<code>borrar_semaforo(int semid)</code>	61
4.17.2.2	<code>crear_semaforo(key_t key, int size, int *semid)</code>	61
4.17.2.3	<code>down_multiple_semaforo(int semid, int size, int undo, int *active)</code>	62
4.17.2.4	<code>down_semaforo(int semid, int num_sem, int undo)</code>	62
4.17.2.5	<code>inicializar_semaforo(int semid, unsigned short *array)</code>	62
4.17.2.6	<code>up_multiple_semaforo(int semid, int size, int undo, int *active)</code>	62
4.17.2.7	<code>up_semaforo(int semid, int num_sem, int undo)</code>	63
4.18	<code>semaforos.h</code> File Reference	63
4.18.1	Detailed Description	64
4.18.2	Macro Definition Documentation	64
4.18.2.1	<code>ERROR</code>	64
4.18.2.2	<code>OK</code>	64
4.18.3	Function Documentation	64
4.18.3.1	<code>borrar_semaforo(int semid)</code>	64
4.18.3.2	<code>crear_semaforo(key_t key, int size, int *semid)</code>	65

4.18.3.3	<code>down_multiple_semaforo(int semid, int size, int undo, int *active)</code>	65
4.18.3.4	<code>down_semaforo(int semid, int num_sem, int undo)</code>	65
4.18.3.5	<code>inicializar_semaforo(int semid, unsigned short *array)</code>	66
4.18.3.6	<code>up_multiple_semaforo(int semid, int size, int undo, int *active)</code>	66
4.18.3.7	<code>up_semaforo(int semid, int num_sem, int undo)</code>	66
4.19	<code>sim_carr_lib.c</code> File Reference	67
4.19.1	Detailed Description	67
4.19.2	Variable Documentation	68
4.19.2.1	<code>din</code>	68
4.19.2.2	<code>n_apos</code>	68
4.19.2.3	<code>n_cab</code>	68
4.19.2.4	<code>n_vent</code>	68
4.19.2.5	<code>num_proc</code>	68
4.20	<code>sim_carr_lib.h</code> File Reference	68
4.20.1	Detailed Description	69
4.20.2	Macro Definition Documentation	70
4.20.2.1	<code>GANADORA</code>	70
4.20.2.2	<code>KEY_APOS_SHM</code>	70
4.20.2.3	<code>KEY_APUES_Q</code>	70
4.20.2.4	<code>KEY_CAB_SEM</code>	70
4.20.2.5	<code>KEY_CAB_SHM</code>	70
4.20.2.6	<code>KEY_GEN_SEM</code>	70
4.20.2.7	<code>KEY_MON_SEM</code>	70
4.20.2.8	<code>KEY_TIR_Q</code>	70
4.20.2.9	<code>KEY_TUR_SEM</code>	70
4.20.2.10	<code>MAX_APOS</code>	70
4.20.2.11	<code>MAX_NAME</code>	71
4.20.2.12	<code>NORMAL</code>	71
4.20.2.13	<code>PATH</code>	71
4.20.2.14	<code>READ</code>	71

4.20.2.15 REMONTAR	71
4.20.2.16 RUTA_FICHERO_APUESTAS	71
4.20.2.17 SIGSTART	71
4.20.2.18 SIGTHROW	71
4.20.2.19 TIEMPO_PRE_CARR	71
4.20.2.20 WRITE	71
4.20.3 Variable Documentation	72
4.20.3.1 din	72
4.20.3.2 n_apos	72
4.20.3.3 n_cab	72
4.20.3.4 n_vent	72
4.20.3.5 num_proc	72
4.21 simular_carreras.c File Reference	72
4.21.1 Detailed Description	73
4.21.2 Macro Definition Documentation	74
4.21.2.1 NUM_ARGS	74
4.21.3 Function Documentation	74
4.21.3.1 _killall(int sig)	74
4.21.3.2 _sim_handler(int sig)	74
4.21.3.3 main(int argc, char *argv[])	74
4.21.3.4 usage()	75
4.21.4 Variable Documentation	75
4.21.4.1 apostadores	75
4.21.4.2 caballos	75
4.21.4.3 gestor	75
4.21.4.4 monitor	75
4.21.4.5 running_principal	75

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_Apostador	Estructura del apostador	5
_Apuesta	Datos de la apuesta	6
_Caballo	Informacion del caballo	7
_ventanilla	Informacion de la ventanilla	9
msgapues	Informacion del mensaje de apuesta	10
msgtir	Estructura del mensaje de tirada	11

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

apostador.c	Implementación de las primitivas del TAD apostador	13
apostador.h	TAD Apostador con su estructura de datos y primitivas	18
apuesta.c	Implementación de las primitivas del TAD apuesta	24
apuesta.h	TAD Apuesta con su estructura y primitivas	28
caballo.c	Implementación de las primitivas del TAD caballo	30
caballo.h	TAD Caballo con su estructura de datos y primitivas	36
mylib.c	Funciones personales varias. Implementacion de mylib.h	43
mylib.h	Interfaz de funciones personales varias	45
rutina_apostador.c	Rutina del proceso apostador	47
rutina_apostador.h	Fichero que contiene la cabecera de la rutina que sigue cada proceso apostador	49
rutina_gestor.c	Rutina del proceso gestor	50
rutina_gestor.h	Fichero que contiene la cabecera de la rutina que sigue el proceso gestor	52
rutina_monitor.c	Rutina del proceso monitor	53
rutina_monitor.h	Fichero que contiene la cabecera de la rutina que sigue el proceso monitor	56
rutina_tirada.c	Rutina del proceso tirada	57
rutina_tirada.h	Fichero que contiene la cabecera de la rutina que sigue cada proceso tirada	59
semaforos.c	Funciones personales acerca de semaforos. Implementacion	60
semaforos.h	Interfaz de funciones personales acerca de semaforos	63

sim_carr_lib.c	
Definicion de las variables externas de sim_carr_lib,h	67
sim_carr_lib.h	
Macros, variables globales y estructuras generales del proyecto	68
simular_carreras.c	
Rutina que simula la carrera	72

Chapter 3

Class Documentation

3.1 `_Apostador` Struct Reference

Estructura del apostador.

```
#include <apostador.h>
```

Public Attributes

- `pid_t` `pid`
- `char` `nombre` [`MAX_APOS_NAME`]
- `double` `ben` [`MAX_CAB`]
- `double` `total`
- `double` `din_rest`

3.1.1 Detailed Description

Estructura del apostador.

Estructura que contiene la informacion del apostador.

3.1.2 Member Data Documentation

3.1.2.1 `double _Apostador::ben[MAX_CAB]`

Beneficios del apostador por cada caballo

3.1.2.2 `double _Apostador::din_rest`

Dinero restante para apostar del apostador

3.1.2.3 char _Apostador::nombre[MAX_APOS_NAME]

Nombre del apostador

3.1.2.4 pid_t _Apostador::pid

Pid del apostador

3.1.2.5 double _Apostador::total

Beneficios totales del apostador

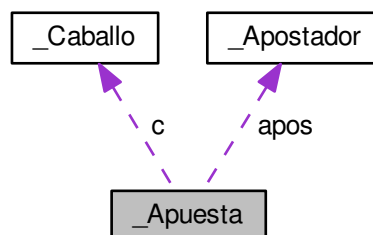
The documentation for this struct was generated from the following file:

- [apostador.h](#)

3.2 _Apuesta Struct Reference

Datos de la apuesta.

Collaboration diagram for _Apuesta:



Public Attributes

- [Apostador](#) * `apos`
- [Caballo](#) * `c`
- int `cab_id`
- unsigned short `ventanilla`
- double `cantidad`

3.2.1 Detailed Description

Datos de la apuesta.

Esta estructura guarda la información de una apuesta.

3.2.2 Member Data Documentation

3.2.2.1 Apostador* _Apuesta::apos

Apostador que realiza la apuesta

3.2.2.2 Caballo* _Apuesta::c

Caballo por el que se apuesta

3.2.2.3 int _Apuesta::cab_id

Identificador del caballo

3.2.2.4 double _Apuesta::cantidad

Cantidad apostada

3.2.2.5 unsigned short _Apuesta::ventanilla

Id de la ventanilla asignada a la apuesta

The documentation for this struct was generated from the following file:

- [apuesta.c](#)

3.3 _Caballo Struct Reference

Informacion del caballo.

```
#include <caballo.h>
```

Public Attributes

- pid_t [pid](#)
- unsigned short [id](#)
- unsigned short [last_tir](#)
- double [cotizacion](#)
- unsigned int [posicion](#)
- double [dinero_apostado](#)

3.3.1 Detailed Description

Informacion del caballo.

Estructura que contiene la informacion del caballo

3.3.2 Member Data Documentation

3.3.2.1 `double _Caballo::cotizacion`

Cotización actual del caballo

3.3.2.2 `double _Caballo::dinero_apostado`

Dinero apostado al caballo

3.3.2.3 `unsigned short _Caballo::id`

Id del caballo

3.3.2.4 `unsigned short _Caballo::last_tir`

Ultima tirada del caballo

3.3.2.5 `pid_t _Caballo::pid`

Pid del caballo

3.3.2.6 `unsigned int _Caballo::posicion`

Posicion actual del caballo

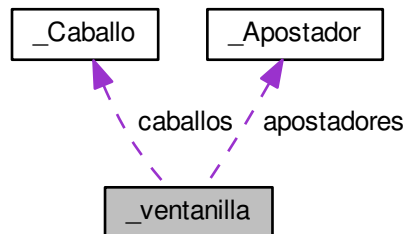
The documentation for this struct was generated from the following file:

- [caballo.h](#)

3.4 _ventanilla Struct Reference

Informacion de la ventana.

Collaboration diagram for _ventanilla:



Public Attributes

- [Caballo](#) * `caballos`
- [Apostador](#) * `apostadores`
- `int` `ventanilla`
- `int` `msgqid`

3.4.1 Detailed Description

Informacion de la ventana.

Esta estructura guarda la informacion que se le envia a cada hilo ventana.

3.4.2 Member Data Documentation

3.4.2.1 `Apostador*` `_ventanilla::apostadores`

Todos los caballos registrados

3.4.2.2 `Caballo*` `_ventanilla::caballos`

Todos los caballos registrados

3.4.2.3 `int` `_ventanilla::msgqid`

Id de la cola de mensajes

3.4.2.4 int _ventanilla::ventanilla

Id de la ventanilla

The documentation for this struct was generated from the following file:

- [rutina_gestor.c](#)

3.5 msgapues Struct Reference

Informacion del mensaje de apuesta.

```
#include <sim_carr_lib.h>
```

Public Attributes

- long [mtype](#)
- char [nombre](#) [[MAX_NAME](#)]
- int [caballo](#)
- double [cantidad](#)

3.5.1 Detailed Description

Informacion del mensaje de apuesta.

Esta estructura guarda la informacion del mensaje de apuesta.

3.5.2 Member Data Documentation

3.5.2.1 int msgapues::caballo

Id del caballo al que apostar

3.5.2.2 double msgapues::cantidad

Cantidad que se apostó

3.5.2.3 long msgapues::mtype

Tipo de mensaje. Es igual al ID del apostador

3.5.2.4 char msgapues::nombre[MAX_NAME]

Nombre del apostador

The documentation for this struct was generated from the following file:

- [sim_carr_lib.h](#)

3.6 msgtir Struct Reference

Estructura del mensaje de tirada.

```
#include <sim_carr_lib.h>
```

Public Attributes

- long [mtype](#)
- unsigned short [tirada](#)

3.6.1 Detailed Description

Estructura del mensaje de tirada.

Esta estructura guarda la informacion del mensaje de tirada.

3.6.2 Member Data Documentation

3.6.2.1 long msgtir::mtype

Tipo del mensaje. Es igual al ID del caballo

3.6.2.2 unsigned short msgtir::tirada

Resultado de la tirada

The documentation for this struct was generated from the following file:

- [sim_carr_lib.h](#)

Chapter 4

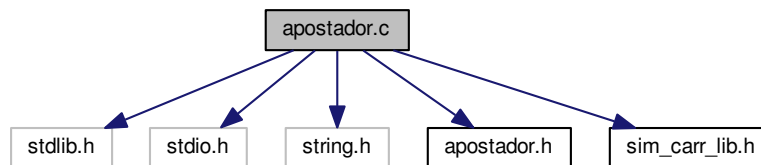
File Documentation

4.1 apostador.c File Reference

Implementación de las primitivas del TAD apostador.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "apostador.h"
#include "sim_carr_lib.h"
```

Include dependency graph for apostador.c:



Functions

- `Apostador * apos_new ()`
Constructor de un apostador.
- `void apos_destroy (Apostador *a)`
Elimina un apostador.
- `Apostador * apos_init (Apostador *a, char *name, double din_init)`
Inicializa los datos de un apostador.
- `Apostador * apos_set_ben (Apostador *a, double beneficio, int id)`
Setter del beneficio del apostador para un cierto caballo.
- `Apostador * apos_set_pid (Apostador *a, pid_t pid)`
Setter del pid de un apostador.
- `char * apos_get_name (Apostador *a)`

- Getter del nombre del apostador.*
- pid_t `apos_get_pid` (`Apostador *a`)
- Getter del pid de un apostador.*
- double `apos_get_ben` (`Apostador *a`, int id)
- Getter del beneficio de un apostador para cierto caballo.*
- `Apostador * apos_incr_din_rest` (`Apostador *a`, double delta)
- Incrementa el dinero disponible del apostador.*
- `Apostador * apos_refresh_total` (`Apostador *a`)
- Refresca los beneficios totales de un apostador.*
- double `apos_get_total` (`Apostador *a`)
- Getter de los beneficios totales del apostador.*
- int `apos_cmp_ben` (const void *v1, const void *v2)
- Compara los beneficios totales de dos apostadores.*

4.1.1 Detailed Description

Implementación de las primitivas del TAD apostador.

Este fichero contiene el código fuente del TAD apostador

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.1.2 Function Documentation

4.1.2.1 int apos_cmp_ben (const void * a1, const void * a2)

Compara los beneficios totales de dos apostadores.

Devuelve la diferencia entre los beneficios totales de dos apostadores

Parameters

<i>a1</i>	Primer apostador
<i>a2</i>	Segundo apostador

Returns

Resta del beneficio del segundo con el beneficio del primero

4.1.2.2 void apos_destroy (Apostador * a)

Elimina un apostador.

Libera la memoria asignada al apostador dado

Parameters

<i>a</i>	Apostador que queremos eliminar
----------	---------------------------------

4.1.2.3 double apos_get_ben (Apostador * a, int id)

Getter del beneficio de un apostador para cierto caballo.

Devuelve el beneficio de un apostador para cierto caballo

Parameters

<i>a</i>	Apostador del que queremos saber el beneficio
<i>id</i>	Id del caballo

Returns

Beneficio del apostador para ese caballo

4.1.2.4 char* apos_get_name (Apostador * a)

Getter del nombre del apostador.

Devuelve el nombre del apostador

Parameters

<i>a</i>	Apostador del que queremos saber el nombre
----------	--

Returns

Nombre del apostador

4.1.2.5 pid_t apos_get_pid (Apostador * a)

Getter del pid de un apostador.

Devuelve el pid del apostador

Parameters

<i>a</i>	Apostador del que queremos saber el pid
----------	---

Returns

Pid del apostador

4.1.2.6 double apos_get_total (Apostador * a)

Getter de los beneficios totales del apostador.

Devuelve los beneficios totales del apostador

Parameters

<i>a</i>	Apostador del que queremos saber los beneficios
----------	---

Returns

Beneficio total del apostador

4.1.2.7 Apostador* apos_incr_din_rest (Apostador * a, double din)

Incrementa el dinero disponible del apostador.

Incrementa el dinero del apostador una cierta cantidad

Parameters

<i>a</i>	Apostador al que incrementar el dinero disponible
<i>din</i>	Dinero que incrementar

Returns

Apostador al que le hemos incrementado el dinero

4.1.2.8 Apostador* apos_init (Apostador * a, char * nombre, double din)

Inicializa los datos de un apostador.

Da nombre y dinero inicial al apostador dado

Parameters

<i>a</i>	Apostador al que queremos poner los datos
<i>nombre</i>	String que representa el nombre del apostador
<i>din</i>	Dinero inicial del apostador

Returns

Apostador inicializado

4.1.2.9 Apostador* apos_new ()

Constructor de un apostador.

Crea un nuevo apostador

Returns

Apostador generado

4.1.2.10 Apostador* apos_refresh_total (Apostador * a)

Refresca los beneficios totales de un apostador.

Actualiza los beneficios totales de un apostador mirando sus beneficios por caballo

Parameters

<i>a</i>	Apostador al que queremos actualizar los beneficios
----------	---

Returns

Apostador al que hemos actualizado los beneficios

4.1.2.11 Apostador* apos_set_ben (Apostador * a, double ben, int id)

Setter del beneficio del apostador para un cierto caballo.

Nos permite establecer el beneficio generado por un cierto caballo al apostador

Parameters

<i>a</i>	Apostador al que queremos establecer el beneficio
<i>ben</i>	Beneficio del apostador
<i>id</i>	Id del caballo del que proviene el beneficio

Returns

Apostador al que le hemos establecido el beneficio de un caballo

4.1.2.12 Apostador* apos_set_pid (Apostador * a, pid_t id)

Setter del pid de un apostador.

Nos permite establecer el pid a un apostador

Parameters

<i>a</i>	Apostador al que queremos establecer el pid
<i>id</i>	Pid del apostador

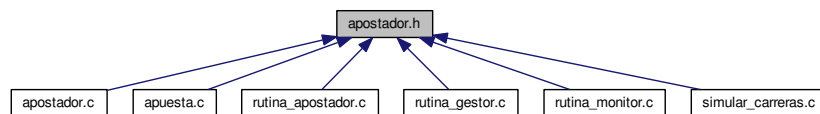
Returns

Apostador al que le hemos establecido el pid

4.2 apostador.h File Reference

TAD Apostador con su estructura de datos y primitivas.

This graph shows which files directly or indirectly include this file:



Classes

- [struct `_Apostador`](#)
Estructura del apostador.

Macros

- `#define` [MAX_APOS_NAME](#) 20
- `#define` [MAX_CAB](#) 10
- `#define` [APOS_ERROR](#) -1

Typedefs

- `typedef struct _Apostador Apostador`
Estructura del apostador.

Functions

- `Apostador * apos_new ()`
Constructor de un apostador.
- `void apos_destroy (Apostador *a)`
Elimina un apostador.
- `Apostador * apos_init (Apostador *a, char *nombre, double din)`
Inicializa los datos de un apostador.
- `Apostador * apos_set_ben (Apostador *a, double ben, int id)`
Setter del beneficio del apostador para un cierto caballo.
- `Apostador * apos_set_pid (Apostador *a, pid_t id)`
Setter del pid de un apostador.
- `Apostador * apos_incr_din_rest (Apostador *a, double din)`
Incrementa el dinero disponible del apostador.
- `char * apos_get_name (Apostador *a)`
Getter del nombre del apostador.
- `double apos_get_ben (Apostador *a, int id)`
Getter del beneficio de un apostador para cierto caballo.
- `pid_t apos_get_pid (Apostador *a)`
Getter del pid de un apostador.
- `double apos_get_total (Apostador *a)`
Getter de los beneficios totales del apostador.
- `int apos_cmp_ben (const void *a1, const void *a2)`
Compara los beneficios totales de dos apostadores.
- `Apostador * apos_refresh_total (Apostador *a)`
Refresca los beneficios totales de un apostador.

4.2.1 Detailed Description

TAD Apostador con su estructura de datos y primitivas.

Este fichero contiene la estructura y funciones necesarias para simular el comportamiento de un apostador.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.2.2 Macro Definition Documentation

4.2.2.1 `#define APOS_ERROR -1`

Codigo de error del apostador

4.2.2.2 `#define MAX_APOS_NAME 20`

Tamaño maximo del nombre del apostador

4.2.2.3 `#define MAX_CAB 10`

Numero maximo de caballos

4.2.3 Typedef Documentation

4.2.3.1 `typedef struct _Apostador Apostador`

Estructura del apostador.

Estructura que contiene la informacion del apostador.

4.2.4 Function Documentation

4.2.4.1 `int apos_cmp_ben (const void * a1, const void * a2)`

Compara los beneficios totales de dos apostadores.

Devuelve la diferencia entre los beneficios totales de dos apostadores

Parameters

<i>a1</i>	Primer apostador
<i>a2</i>	Segundo apostador

Returns

Resta del beneficio del segundo con el beneficio del primero

4.2.4.2 `void apos_destroy (Apostador * a)`

Elimina un apostador.

Libera la memoria asignada al apostador dado

Parameters

<i>a</i>	Apostador que queremos eliminar
----------	---------------------------------

4.2.4.3 double apos_get_ben (Apostador * a, int id)

Getter del beneficio de un apostador para cierto caballo.

Devuelve el beneficio de un apostador para cierto caballo

Parameters

<i>a</i>	Apostador del que queremos saber el beneficio
<i>id</i>	Id del caballo

Returns

Beneficio del apostador para ese caballo

4.2.4.4 char* apos_get_name (Apostador * a)

Getter del nombre del apostador.

Devuelve el nombre del apostador

Parameters

<i>a</i>	Apostador del que queremos saber el nombre
----------	--

Returns

Nombre del apostador

4.2.4.5 pid_t apos_get_pid (Apostador * a)

Getter del pid de un apostador.

Devuelve el pid del apostador

Parameters

<i>a</i>	Apostador del que queremos saber el pid
----------	---

Returns

Pid del apostador

4.2.4.6 double apos_get_total (Apostador * a)

Getter de los beneficios totales del apostador.

Devuelve los beneficios totales del apostador

Parameters

<i>a</i>	Apostador del que queremos saber los beneficios
----------	---

Returns

Beneficio total del apostador

4.2.4.7 Apostador* apos_incr_din_rest (Apostador * *a*, double *din*)

Incrementa el dinero disponible del apostador.

Incrementa el dinero del apostador una cierta cantidad

Parameters

<i>a</i>	Apostador al que incrementar el dinero disponible
<i>din</i>	Dinero que incrementar

Returns

Apostador al que le hemos incrementado el dinero

4.2.4.8 Apostador* apos_init (Apostador * *a*, char * *nombre*, double *din*)

Inicializa los datos de un apostador.

Da nombre y dinero inicial al apostador dado

Parameters

<i>a</i>	Apostador al que queremos poner los datos
<i>nombre</i>	String que representa el nombre del apostador
<i>din</i>	Dinero inicial del apostador

Returns

Apostador inicializado

4.2.4.9 Apostador* apos_new ()

Constructor de un apostador.

Crea un nuevo apostador

Returns

Apostador generado

4.2.4.10 Apostador* apos_refresh_total (Apostador * a)

Refresca los beneficios totales de un apostador.

Actualiza los beneficios totales de un apostador mirando sus beneficios por caballo

Parameters

<i>a</i>	Apostador al que queremos actualizar los beneficios
----------	---

Returns

Apostador al que hemos actualizado los beneficios

4.2.4.11 Apostador* apos_set_ben (Apostador * a, double ben, int id)

Setter del beneficio del apostador para un cierto caballo.

Nos permite establecer el beneficio generado por un cierto caballo al apostador

Parameters

<i>a</i>	Apostador al que queremos establecer el beneficio
<i>ben</i>	Beneficio del apostador
<i>id</i>	Id del caballo del que proviene el beneficio

Returns

Apostador al que le hemos establecido el beneficio de un caballo

4.2.4.12 Apostador* apos_set_pid (Apostador * a, pid_t id)

Setter del pid de un apostador.

Nos permite establecer el pid a un apostador

Parameters

<i>a</i>	Apostador al que queremos establecer el pid
<i>id</i>	Pid del apostador

Returns

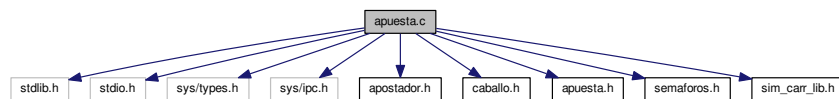
Apostador al que le hemos establecido el pid

4.3 apuesta.c File Reference

Implementación de las primitivas del TAD apuesta.

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include "apostador.h"
#include "caballo.h"
#include "apuesta.h"
#include "semaforos.h"
#include "sim_carr_lib.h"
```

Include dependency graph for apuesta.c:



Classes

- [struct _Apuesta](#)

Datos de la apuesta.

Functions

- [void _apuesta_print](#) (FILE *fp, [Apuesta](#) *a, double old_cot)
Función privada que guarda los datos de una apuesta.
- [Apuesta](#) * [apuesta_new](#) ()
Constructor de una apuesta.
- [void](#) [apuesta_destroy](#) ([Apuesta](#) *a)
Destructor de una apuesta.
- [Apuesta](#) * [apuesta_init](#) ([Apuesta](#) *a, [Apostador](#) *apos, [Caballo](#) *c, int cab_id, unsigned short ventanilla, double cantidad)
Inicializador de los valores de la apuesta.
- [void](#) [apuesta_execute](#) ([Apuesta](#) *a, char *path)
Ejecuta una apuesta.

Variables

- double [apuesta_total](#) = 0

4.3.1 Detailed Description

Implementación de las primitivas del TAD apuesta.

Este fichero contiene el código fuente del TAD apuesta

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.3.2 Function Documentation

4.3.2.1 void _apuesta_print (FILE * *fp*, Apuesta * *a*, double *old_cot*)

Función privada que guarda los datos de una apuesta.

Imprime los datos de la apuesta en un fichero dado

Parameters

<i>fp</i>	Fichero en el que escribimos los datos
<i>a</i>	Apuesta de la que queremos guardar los datos
<i>old_cot</i>	Cotización anterior del caballo

4.3.2.2 void apuesta_destroy (Apuesta * *a*)

Destructor de una apuesta.

Elimina una apuesta dada, liberando su memoria

Parameters

<i>a</i>	Apuesta a eliminar
----------	--------------------

4.3.2.3 void apuesta_execute (Apuesta * *a*, char * *filepath*)

Ejecuta una apuesta.

Ejecuta la apuesta a. Actualiza la cotizacion de los caballos, el estado del apostador y escribe en un fichero dado por filepath el registro de la apuesta

Parameters

<i>a</i>	Apuesta a ejecutar
<i>filepath</i>	Ruta del fichero donde guardar las apuestas

4.3.2.4 `Apuesta* apuesta_init (Apuesta * a, Apostador * apos, Caballo * c, int cab_id, unsigned short ventanilla, double cantidad)`

Inicializador de los valores de la apuesta.

Establece los valores iniciales de una apuesta dada

Parameters

<i>a</i>	Apuesta a la que inicializar los valores
<i>apos</i>	Apostador que ha generado la apuesta
<i>c</i>	Caballo de la apuesta
<i>cab_id</i>	Identificador del caballo de la apuesta
<i>ventanilla</i>	Identificador de la ventanilla donde se gestionará la apuesta
<i>cantidad</i>	Dinero apostado

Returns

Apuesta que hemos inicializado

4.3.2.5 `Apuesta* apuesta_new ()`

Constructor de una apuesta.

Crea una nueva apuesta, reservando memoria para ella

Returns

Apuesta creada

4.3.3 Variable Documentation

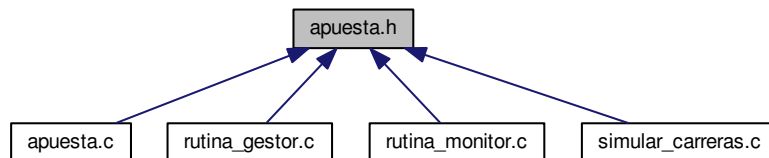
4.3.3.1 `double apuesta_total = 0`

Variable que contiene el total de todas las apuestas realizadas hasta el momento

4.4 apuesta.h File Reference

TAD Apuesta con su estructura y primitivas.

This graph shows which files directly or indirectly include this file:



Typedefs

- typedef struct [_Apuesta](#) [Apuesta](#)

Functions

- [Apuesta](#) * [apuesta_new](#) ()
Constructor de una apuesta.
- void [apuesta_destroy](#) ([Apuesta](#) *a)
Destructor de una apuesta.
- [Apuesta](#) * [apuesta_init](#) ([Apuesta](#) *a, [Apostador](#) *apos, [Caballo](#) *c, int cab_id, unsigned short ventanilla, double cantidad)
Inicializador de los valores de la apuesta.
- void [apuesta_execute](#) ([Apuesta](#) *a, char *filepath)
Ejecuta una apuesta.

4.4.1 Detailed Description

TAD Apuesta con su estructura y primitivas.

Este fichero contiene la estructura y funciones necesarias para simular el comportamiento de un apostador.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.4.2 Typedef Documentation

4.4.2.1 typedef struct _Apuesta Apuesta

Define el tipo de datos Apuesta como la estructura de la misma

4.4.3 Function Documentation

4.4.3.1 void apuesta_destroy (Apuesta * a)

Destructor de una apuesta.

Elimina una apuesta dada, liberando su memoria

Parameters

<i>a</i>	Apuesta a eliminar
----------	--------------------

4.4.3.2 void apuesta_execute (Apuesta * a, char * filepath)

Ejecuta una apuesta.

Ejecuta la apuesta a. Actualiza la cotizacion de los caballos, el estado del apostador y escribe en un fichero dado por filepath el registro de la apuesta

Parameters

<i>a</i>	Apuesta a ejecutar
<i>filepath</i>	Ruta del fichero donde guardar las apuestas

4.4.3.3 Apuesta* apuesta_init (Apuesta * a, Apostador * apos, Caballo * c, int cab_id, unsigned short ventanilla, double cantidad)

Inicializador de los valores de la apuesta.

Establece los valores iniciales de una apuesta dada

Parameters

<i>a</i>	Apuesta a la que inicializar los valores
<i>apos</i>	Apostador que ha generado la apuesta
<i>c</i>	Caballo de la apuesta
<i>cab_id</i>	Identificador del caballo de la apuesta
<i>ventanilla</i>	Identificador de la ventanilla donde se gestionará la apuesta
<i>cantidad</i>	Dinero apostado

Returns

Apuesta que hemos inicializado

4.4.3.4 Apuesta* apuesta_new ()

Constructor de una apuesta.

Crea una nueva apuesta, reservando memoria para ella

Returns

Apuesta creada

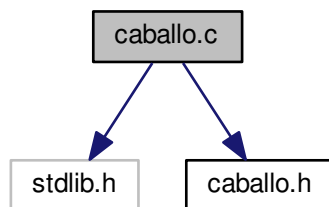
4.5 caballo.c File Reference

Implementación de las primitivas del TAD caballo.

```
#include <stdlib.h>
```

```
#include "caballo.h"
```

Include dependency graph for caballo.c:

**Functions**

- `Caballo * cab_new ()`
Constructor de un caballo.
- `void cab_destroy (Caballo *c)`
Elimina un caballo.
- `Caballo * cab_set_id (Caballo *c, unsigned short id)`
Setter del id del caballo.
- `Caballo * cab_set_cot (Caballo *c, double cotizacion)`
Setter de la cotización del caballo.
- `Caballo * cab_set_pos (Caballo *c, unsigned int posicion)`
Setter de la posición del caballo.
- `Caballo * cab_set_pid (Caballo *c, pid_t pid)`
Setter del pid del caballo.

- `Caballo * cab_incr_apostado (Caballo *c, double delta)`
Incrementa el dinero apostado al caballo.
- `Caballo * cab_set_last_tir (Caballo *c, unsigned short tir)`
Setter de la ultima tirada del caballo.
- `unsigned short cab_get_last_tir (Caballo *c)`
Getter de la ultima tirada del caballo.
- `unsigned short cab_get_id (Caballo *c)`
Getter del id del caballo.
- `unsigned int cab_get_pos (Caballo *c)`
Getter de la posicion del caballo.
- `pid_t cab_get_pid (Caballo *c)`
Getter del pid del caballo.
- `double cab_get_cot (Caballo *c)`
Getter de la cotizacion del caballo.
- `double cab_get_apostado (Caballo *c)`
Getter del dinero apostado del caballo.

4.5.1 Detailed Description

Implementación de las primitivas del TAD caballo.

Este fichero contiene el código fuente del TAD caballo

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.5.2 Function Documentation

4.5.2.1 void cab_destroy (Caballo * c)

Elimina un caballo.

Libera la memoria asignada al caballo dado

Parameters

<code>c</code>	Caballo que queremos eliminar
----------------	-------------------------------

4.5.2.2 double cab_get_apostado (Caballo * c)

Getter del dinero apostado del caballo.

Devuelve el dinero apostado del caballo

Parameters

c	Caballo del que se devuelve el dinero apostado
---	--

Returns

Dinero apostado

4.5.2.3 double cab_get_cot (Caballo * c)

Getter de la cotizacion del caballo.

Devuelve la cotizacion del caballo

Parameters

c	Caballo del que se devuelve la cotizacion
---	---

Returns

Cotizacion del caballo

4.5.2.4 unsigned short cab_get_id (Caballo * c)

Getter del id del caballo.

Devuelve el id del caballo

Parameters

c	Caballo del que se devuelve el id
---	-----------------------------------

Returns

Id del caballo

4.5.2.5 unsigned short cab_get_last_tir (Caballo * c)

Getter de la ultima tirada del caballo.

Devuelve la ultima tirada del caballo

Parameters

<i>c</i>	Caballo del que se devuelve la ultima tirada
----------	--

Returns

Ultima tirada del caballo

4.5.2.6 `pid_t cab_get_pid (Caballo * c)`

Getter del pid del caballo.

Devuelve el pid del caballo

Parameters

<i>c</i>	Caballo del que se devuelve el pid
----------	------------------------------------

Returns

Pid del caballo

4.5.2.7 `unsigned int cab_get_pos (Caballo * c)`

Getter de la posicion del caballo.

Devuelve la posicion del caballo

Parameters

<i>c</i>	Caballo del que se devuelve la posicion
----------	---

Returns

Posicion del caballo

4.5.2.8 `Caballo* cab_incr_apostado (Caballo * c, double incr)`

Incrementa el dinero apostado al caballo.

Incrementa el dinero apostado al caballo

Parameters

<i>c</i>	Caballo a modificar el dinero apostado
<i>incr</i>	Incremento a añadir

Returns

Caballo al que hemos cambiado el dinero apostado

4.5.2.9 Caballo* cab_new ()

Constructor de un caballo.

Crea un nuevo caballo

Returns

Caballo generado

4.5.2.10 Caballo* cab_set_cot (Caballo * c, double cot)

Setter de la cotización del caballo.

Establece la cotización del caballo

Parameters

<i>c</i>	Caballo a cambiar la cotización
<i>cot</i>	Cotización del caballo

Returns

Caballo al que hemos cambiado la cotización

4.5.2.11 Caballo* cab_set_id (Caballo * c, unsigned short id)

Setter del id del caballo.

Establece el id del caballo

Parameters

<i>c</i>	Caballo a cambiar el id
<i>id</i>	Id del caballo

Returns

Caballo al que hemos cambiado el id

4.5.2.12 Caballo* cab_set_last_tir (Caballo * c, unsigned short tir)

Setter de la ultima tirada del caballo.

Establece la ultima tirada del caballo

Parameters

<i>c</i>	Caballo a cambiar la tirada
<i>tir</i>	Valor de la ultima tirada del caballo

Returns

Caballo al que hemos cambiado la ultima tirada

4.5.2.13 Caballo* cab_set_pid (Caballo * c, pid_t pid)

Setter del pid del caballo.

Establece el pid del caballo

Parameters

<i>c</i>	Caballo a cambiar el pid
<i>pid</i>	Pid del caballo

Returns

Caballo al que hemos cambiado el id

4.5.2.14 Caballo* cab_set_pos (Caballo * c, unsigned int pos)

Setter de la posición del caballo.

Establece la posición del caballo

Parameters

<i>c</i>	Caballo a cambiar la posicion
<i>pos</i>	Posición del caballo

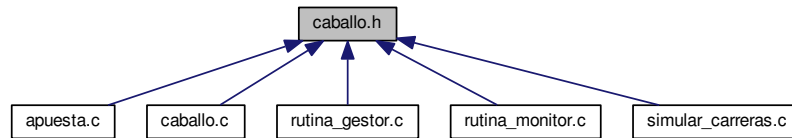
Returns

Caballo al que hemos cambiado la posición

4.6 caballo.h File Reference

TAD Caballo con su estructura de datos y primitivas.

This graph shows which files directly or indirectly include this file:



Classes

- struct [_Caballo](#)
Informacion del caballo.

Macros

- #define [CAB_ERROR](#) -1;

Typedefs

- typedef struct [_Caballo](#) Caballo
Informacion del caballo.

Functions

- Caballo * [cab_new](#) ()
Constructor de un caballo.
- void [cab_destroy](#) (Caballo *c)
Elimina un caballo.
- Caballo * [cab_set_cot](#) (Caballo *c, double cot)
Setter de la cotización del caballo.
- Caballo * [cab_set_pos](#) (Caballo *c, unsigned int pos)
Setter de la posición del caballo.
- Caballo * [cab_set_id](#) (Caballo *c, unsigned short id)
Setter del id del caballo.
- Caballo * [cab_set_pid](#) (Caballo *c, pid_t pid)
Setter del pid del caballo.
- Caballo * [cab_set_last_tir](#) (Caballo *c, unsigned short tir)
Setter de la ultima tirada del caballo.
- Caballo * [cab_incr_apostado](#) (Caballo *c, double incr)
Incrementa el dinero apostado al caballo.
- unsigned short [cab_get_id](#) (Caballo *c)
Getter del id del caballo.
- unsigned short [cab_get_last_tir](#) (Caballo *c)
Getter de la ultima tirada del caballo.
- unsigned int [cab_get_pos](#) (Caballo *c)

- Getter de la posicion del caballo.*
 - pid_t `cab_get_pid` (`Caballo *c`)
Getter del pid del caballo.
 - double `cab_get_cot` (`Caballo *c`)
Getter de la cotizacion del caballo.
 - double `cab_get_apostado` (`Caballo *c`)
Getter del dinero apostado del caballo.

4.6.1 Detailed Description

TAD Caballo con su estructura de datos y primitivas.

Este fichero contiene la estructura y funciones necesarias para simular el comportamiento de un caballo.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.6.2 Macro Definition Documentation

4.6.2.1 #define CAB_ERROR -1;

Codigo de error del caballo

4.6.3 Typedef Documentation

4.6.3.1 typedef struct _Caballo Caballo

Informacion del caballo.

Estructura que contiene la informacion del caballo

4.6.4 Function Documentation

4.6.4.1 void cab_destroy (Caballo * c)

Elimina un caballo.

Libera la memoria asignada al caballo dado

Parameters

<i>c</i>	Caballo que queremos eliminar
----------	-------------------------------

4.6.4.2 double cab_get_apostado (**Caballo** * *c*)

Getter del dinero apostado del caballo.

Devuelve el dinero apostado del caballo

Parameters

<i>c</i>	Caballo del que se devuelve el dinero apostado
----------	--

Returns

Dinero apostado

4.6.4.3 double cab_get_cot (**Caballo** * *c*)

Getter de la cotizacion del caballo.

Devuelve la cotizacion del caballo

Parameters

<i>c</i>	Caballo del que se devuelve la cotizacion
----------	---

Returns

Cotizacion del caballo

4.6.4.4 unsigned short cab_get_id (**Caballo** * *c*)

Getter del id del caballo.

Devuelve el id del caballo

Parameters

<i>c</i>	Caballo del que se devuelve el id
----------	-----------------------------------

Returns

Id del caballo

4.6.4.5 unsigned short cab_get_last_tir (Caballo * c)

Getter de la ultima tirada del caballo.

Devuelve la ultima tirada del caballo

Parameters

<i>c</i>	Caballo del que se devuelve la ultima tirada
----------	--

Returns

Ultima tirada del caballo

4.6.4.6 pid_t cab_get_pid (Caballo * c)

Getter del pid del caballo.

Devuelve el pid del caballo

Parameters

<i>c</i>	Caballo del que se devuelve el pid
----------	------------------------------------

Returns

Pid del caballo

4.6.4.7 unsigned int cab_get_pos (Caballo * c)

Getter de la posicion del caballo.

Devuelve la posicion del caballo

Parameters

<i>c</i>	Caballo del que se devuelve la posicion
----------	---

Returns

Posicion del caballo

4.6.4.8 Caballo* cab_incr_apostado (Caballo * c, double *incr*)

Incrementa el dinero apostado al caballo.

Incrementa el dinero apostado al caballo

Parameters

<i>c</i>	Caballo a modificar el dinero apostado
<i>incr</i>	Incremento a añadir

Returns

Caballo al que hemos cambiado el dinero apostado

4.6.4.9 Caballo* cab_new ()

Constructor de un caballo.

Crea un nuevo caballo

Returns

Caballo generado

4.6.4.10 Caballo* cab_set_cot (Caballo * c, double cot)

Setter de la cotización del caballo.

Establece la cotización del caballo

Parameters

<i>c</i>	Caballo a cambiar la cotización
<i>cot</i>	Cotización del caballo

Returns

Caballo al que hemos cambiado la cotización

4.6.4.11 Caballo* cab_set_id (Caballo * c, unsigned short id)

Setter del id del caballo.

Establece el id del caballo

Parameters

<i>c</i>	Caballo a cambiar el id
<i>id</i>	Id del caballo

Returns

Caballo al que hemos cambiado el id

4.6.4.12 Caballo* cab_set_last_tir (Caballo * c, unsigned short tir)

Setter de la ultima tirada del caballo.

Establece la ultima tirada del caballo

Parameters

<i>c</i>	Caballo a cambiar la tirada
<i>tir</i>	Valor de la ultima tirada del caballo

Returns

Caballo al que hemos cambiado la ultima tirada

4.6.4.13 Caballo* cab_set_pid (Caballo * c, pid_t pid)

Setter del pid del caballo.

Establece el pid del caballo

Parameters

<i>c</i>	Caballo a cambiar el pid
<i>pid</i>	Pid del caballo

Returns

Caballo al que hemos cambiado el id

4.6.4.14 Caballo* cab_set_pos (Caballo * c, unsigned int pos)

Setter de la posición del caballo.

Establece la posición del caballo

Parameters

<i>c</i>	Caballo a cambiar la posicion
<i>pos</i>	Posición del caballo

Returns

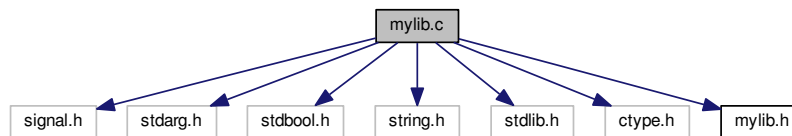
Caballo al que hemos cambiado la posición

4.7 mylib.c File Reference

Funciones personales varias. Implementacion de [mylib.h](#).

```
#include <signal.h>
#include <stdarg.h>
#include <stdbool.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include "mylib.h"
```

Include dependency graph for mylib.c:



Functions

- void [sigaddset_var](#) (sigset_t *sig, int sig,...)
Añade una lista de señales a un set.
- float [randNum](#) (float inf, float sup)
Genera un número real aleatorio en el rango [inf, sup)
- bool [aredigits](#) (const char *string)
Comprueba si una string es numerica.
- bool [isfloat](#) (const char *string)
Comprueba si una string es un float o entero.

4.7.1 Detailed Description

Funciones personales varias. Implementacion de [mylib.h](#).

Este fichero contiene el código fuente de las funciones en la interfaz [mylib.h](#).

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

12-04-2018

4.7.2 Function Documentation

4.7.2.1 bool aredigits (const char * *string*)

Comprueba si una string es numerica.

Comprueba si cada caracter de la cadena string es un valor numerico.

Parameters

<i>string</i>	Cadena de caracteres.
---------------	-----------------------

Returns

Devuelve true si la cadena es numerica, false si no lo es.

4.7.2.2 bool isfloat (const char * *string*)

Comprueba si una string es un float o entero.

Comprueba si cada caracter de la cadena string es un valor numerico o un punto.

Parameters

<i>string</i>	Cadena de caracteres.
---------------	-----------------------

Returns

Devuelve true si la cadena es un float o entero, false si no lo es.

4.7.2.3 float randNum (float *inf*, float *sup*)

Genera un número real aleatorio en el rango [inf, sup)

Genera un número real aleatorio en el rango [inf, sup)

Parameters

<i>inf</i>	Límite inferior del intervalo
<i>sup</i>	Límite superior del intervalo

Returns

Devuelve el número aleatorio generado

4.7.2.4 void sigaddset_var (sigset_t * sigt, int sig, ...)

Añade una lista de señales a un set.

Añade una lista de señales acabada en -1 al set dado por sigt. Ejemplo de uso : sigaddset_var(sigt, SIGUSR1, SIGUSR2, SIGINT, -1);

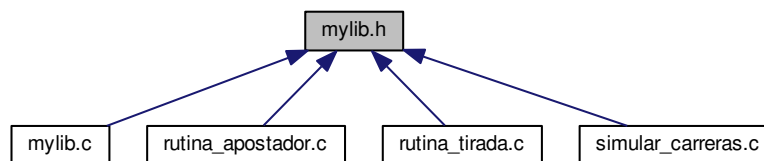
Parameters

<i>sigt</i>	Puntero de tipo sigset_t. Set de señales.
<i>sig</i>	Indices de las distintas señales. De extension variable. Debe terminar en -1.

4.8 mylib.h File Reference

Interfaz de funciones personales varias.

This graph shows which files directly or indirectly include this file:



Functions

- void [sigaddset_var](#) (sigset_t *sigt, int sig,...)
Añade una lista de señales a un set.
- float [randNum](#) (float inf, float sup)
Genera un número real aleatorio en el rango [inf, sup)
- bool [aredigits](#) (const char *string)
Comprueba si una string es numerica.
- bool [isfloat](#) (const char *string)
Comprueba si una string es un float o entero.

4.8.1 Detailed Description

Interfaz de funciones personales varias.

Este fichero contiene las funciones personales varias.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

12-04-2018

4.8.2 Function Documentation

4.8.2.1 `bool aredigits (const char * string)`

Comprueba si una string es numerica.

Comprueba si cada caracter de la cadena string es un valor numerico.

Parameters

<i>string</i>	Cadena de caracteres.
---------------	-----------------------

Returns

Devuelve true si la cadena es numerica, false si no lo es.

4.8.2.2 `bool isfloat (const char * string)`

Comprueba si una string es un float o entero.

Comprueba si cada caracter de la cadena string es un valor numerico o un punto.

Parameters

<i>string</i>	Cadena de caracteres.
---------------	-----------------------

Returns

Devuelve true si la cadena es un float o entero, false si no lo es.

4.8.2.3 `float randNum (float inf, float sup)`

Genera un número real aleatorio en el rango [inf, sup)

Genera un número real aleatorio en el rango [inf, sup)

Parameters

<i>inf</i>	Límite inferior del intervalo
<i>sup</i>	Límite superior del intervalo

Returns

Devuelve el número aleatorio generado

4.8.2.4 void sigaddset_var (sigset_t * sigt, int sig, ...)

Añade una lista de señales a un set.

Añade una lista de señales acabada en -1 al set dado por sigt. Ejemplo de uso : sigaddset_var(sigt, SIGUSR1, SIGUSR2, SIGINT, -1);

Parameters

<i>sigt</i>	Puntero de tipo sigset_t. Set de señales.
<i>sig</i>	Indices de las distintas señales. De extensión variable. Debe terminar en -1.

4.9 rutina_apostador.c File Reference

Rutina del proceso apostador.

```
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sys/shm.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <unistd.h>
#include <stdbool.h>
#include <errno.h>
#include "mylib.h"
#include "apostador.h"
#include "semaforos.h"
#include "sim_carr_lib.h"
#include "rutina_apostador.h"
#include "time.h"
```

Include dependency graph for rutina_apostador.c:



Functions

- void `_apos_handler` (int sig)
Manejador de señales del apostador.
- void `proc_apostador` (int id)
Rutina del proceso apostador.

Variables

- volatile bool `running_apostador` = true
- `Apostador` * `apostadores`

4.9.1 Detailed Description

Rutina del proceso apostador.

Este fichero contiene el código fuente de la simulación del apostador

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.9.2 Function Documentation

4.9.2.1 void _apos_handler (int sig)

Manejador de señales del apostador.

Define las acciones a realizar por el apostador en la recepción de ciertas señales

Parameters

<code>sig</code>	Señal recibida
------------------	----------------

4.9.2.2 void proc_apostador (int id)

Rutina del proceso apostador.

Funcion que ejecuta el proceso apostador nada más haberse bifurcado

Parameters

<i>id</i>	Id del apostador
-----------	------------------

4.9.3 Variable Documentation

4.9.3.1 Apostador* apostadores

Apostadores registrados

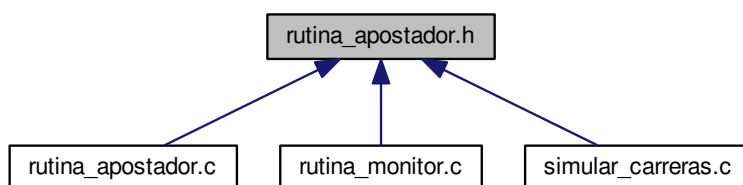
4.9.3.2 volatile bool running_apostador = true

Bandera que indica si el apostador sigue apostando

4.10 rutina_apostador.h File Reference

Fichero que contiene la cabecera de la rutina que sigue cada proceso apostador.

This graph shows which files directly or indirectly include this file:



Functions

- void [proc_apostador](#) (int id)
Rutina del proceso apostador.

4.10.1 Detailed Description

Fichero que contiene la cabecera de la rutina que sigue cada proceso apostador.

Este fichero contiene la cabecera de la rutina que sigue cada proceso apostador.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.10.2 Function Documentation

4.10.2.1 void proc_apostador (int id)

Rutina del proceso apostador.

Funcion que ejecuta el proceso apostador nada más haberse bifurcado

Parameters

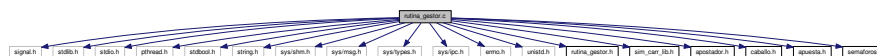
<i>id</i>	Id del apostador
-----------	------------------

4.11 rutina_gestor.c File Reference

Rutina del proceso gestor.

```
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#include <stdbool.h>
#include <string.h>
#include <sys/shm.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <errno.h>
#include <unistd.h>
#include "rutina_gestor.h"
#include "sim_carr_lib.h"
#include "apostador.h"
#include "caballo.h"
#include "apuesta.h"
#include "semaforos.h"
```

Include dependency graph for rutina_gestor.c:



Classes

- struct [_ventanilla](#)
Informacion de la ventanilla.

Functions

- void * [_rutina_ventanilla](#) (void *data)
Función privada del gestor que realiza la recepcion de las apuestas.
- void [_gestor_handler](#) (int sig)
Manejador de señales del proceso gestor.
- void [proc_gestor](#) ()
Rutina del proceso gestor.

Variables

- volatile bool `end` = false

4.11.1 Detailed Description

Rutina del proceso gestor.

Este fichero contiene el código fuente de la simulación del gestor

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.11.2 Function Documentation

4.11.2.1 void _gestor_handler (int *sig*)

Manejador de señales del proceso gestor.

Establece las acciones que deberá ejecutar el proceso cuando reciba ciertas señales

Parameters

<i>sig</i>	Señal recibida
------------	----------------

4.11.2.2 void * _rutina_ventanilla (void * *data*)

Función privada del gestor que realiza la recepcion de las apuestas.

Simula la recepción de apuestas y su gestión

Parameters

<i>data</i>	Puntero a void que contiene toda la información necesaria para ejecutar la rutina
-------------	---

4.11.2.3 void `proc_gestor` ()

Rutina del proceso gestor.

Funcion que ejecuta el proceso gestor nada más haberse bifurcado

4.11.3 Variable Documentation

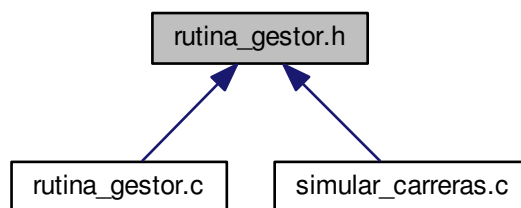
4.11.3.1 volatile bool `end` = false

Booleano que representa si el gestor tiene que acabar o no

4.12 `rutina_gestor.h` File Reference

Fichero que contiene la cabecera de la rutina que sigue el proceso gestor.

This graph shows which files directly or indirectly include this file:



Functions

- void `proc_gestor` ()
Rutina del proceso gestor.

4.12.1 Detailed Description

Fichero que contiene la cabecera de la rutina que sigue el proceso gestor.

Este fichero contiene la cabecera de la rutina que sigue el proceso gestor de apuestas.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.12.2 Function Documentation

4.12.2.1 void proc_gestor ()

Rutina del proceso gestor.

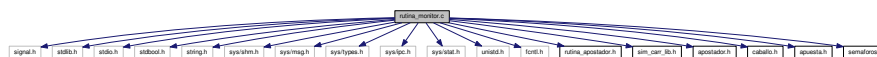
Funcion que ejecuta el proceso gestor nada más haberse bifurcado

4.13 rutina_monitor.c File Reference

Rutina del proceso monitor.

```
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <sys/shm.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/stat.h>
#include <unistd.h>
#include <fcntl.h>
#include "rutina_apostador.h"
#include "sim_carr_lib.h"
#include "apostador.h"
#include "caballo.h"
#include "apuesta.h"
#include "semaforos.h"
```

Include dependency graph for rutina_monitor.c:



Macros

- #define [MAX_APOS_PRINT](#) 10

Functions

- void [_monitor_handler](#) (int sig)
Manejador de señales del proceso monitor.
- void [_monitor_pre_carrera](#) (Caballo *c)
Función auxiliar que define el comportamiento del proceso antes de la carrera.
- void [_monitor_carrera](#) (Caballo *c)
Función auxiliar que define el comportamiento del proceso durante la carrera.
- void [_monitor_post_carrera](#) (Caballo *c, Apostador *a)
Función auxiliar que define el comportamiento del proceso tras la carrera.
- void [_monitor_fin](#) ()
Función auxiliar que define el comportamiento del proceso al acabar la carrera.
- void [proc_monitor](#) ()
Rutina del proceso monitor.

Variables

- volatile bool `fin_pre_carr` = false
- volatile bool `fin_carr` = false

4.13.1 Detailed Description

Rutina del proceso monitor.

Este fichero contiene el código fuente de la simulación del monitor

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.13.2 Macro Definition Documentation

4.13.2.1 `#define MAX_APOS_PRINT 10`

Numero maximo de apostadores que imprimir en el ranking al finalizar la carrera

4.13.3 Function Documentation

4.13.3.1 `void _monitor_carrera (Caballo * c)`

Función auxiliar que define el comportamiento del proceso durante la carrera.

Imprime los datos necesarios durante la carrera

Parameters

<code>c</code>	Caballos registrados
----------------	----------------------

4.13.3.2 `void _monitor_fin ()`

Función auxiliar que define el comportamiento del proceso al acabar la carrera.

Imprime los datos necesarios al acabar la carrera

4.13.3.3 void _monitor_handler (int *sig*)

Manejador de señales del proceso monitor.

Establece las acciones que deberá ejecutar el proceso cuando reciba ciertas señales

Parameters

<i>sig</i>	Señal recibida
------------	----------------

4.13.3.4 void _monitor_post_carrera (**Caballo** * *c*, **Apostador** * *a*)

Función auxiliar que define el comportamiento del proceso tras la carrera.

Imprime los datos necesarios tras la carrera

Parameters

<i>c</i>	Caballos registrados
<i>a</i>	Apostadores registrados

4.13.3.5 void _monitor_pre_carrera (**Caballo** * *c*)

Función auxiliar que define el comportamiento del proceso antes de la carrera.

Imprime los datos necesarios en el tiempo anterior a la carrera

Parameters

<i>c</i>	Caballos registrados
----------	----------------------

4.13.3.6 void proc_monitor ()

Rutina del proceso monitor.

Funcion que ejecuta el proceso monitor nada más haberse bifurcado

4.13.4 Variable Documentation

4.13.4.1 volatile bool fin_carr = false

Flag que indica el fin de la carrera

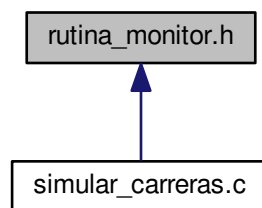
4.13.4.2 volatile bool fin_pre_carr = false

Flag que indica el fin del tiempo de pre carrera

4.14 rutina_monitor.h File Reference

Fichero que contiene la cabecera de la rutina que sigue el proceso monitor.

This graph shows which files directly or indirectly include this file:



Functions

- void [proc_monitor](#) ()
Rutina del proceso monitor.

4.14.1 Detailed Description

Fichero que contiene la cabecera de la rutina que sigue el proceso monitor.

Este fichero contiene la cabecera de la rutina que sigue el proceso monitor.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.14.2 Function Documentation

4.14.2.1 void proc_monitor ()

Rutina del proceso monitor.

Funcion que ejecuta el proceso monitor nada más haberse bifurcado

4.15 rutina_tirada.c File Reference

Rutina del proceso tirada.

```
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <stdbool.h>
#include <time.h>
#include "sim_carr_lib.h"
#include "mylib.h"
#include "semaforos.h"
#include "rutina_tirada.h"
```

Include dependency graph for rutina_tirada.c:



Functions

- void [_tirada_handler](#) (int sig)
Manejador de señales del proceso tirada.
- void [proc_tirada](#) (int id, int *pipe)
Rutina del proceso monitor.

Variables

- volatile bool [running_tirada](#) = true

4.15.1 Detailed Description

Rutina del proceso tirada.

Este fichero contiene el código fuente de la simulación de la tirada

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.15.2 Function Documentation

4.15.2.1 void_tirada_handler (int *sig*)

Manejador de señales del proceso tirada.

Establece las acciones que deberá ejecutar el proceso cuando reciba ciertas señales

Parameters

<i>sig</i>	Señal recibida
------------	----------------

4.15.2.2 void proc_tirada (int *id*, int * *pipe*)

Rutina del proceso monitor.

Funcion que ejecuta el proceso monitor nada más haberse bifurcado

Parameters

<i>id</i>	Id del caballo que realiza las tiradas
<i>pipe</i>	Tubería por la que se comunica el tipo de tirada al caballo.

4.15.3 Variable Documentation

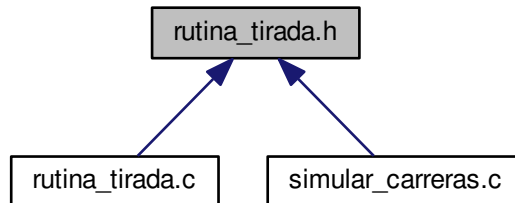
4.15.3.1 volatile bool running_tirada = true

Flag que indica si hay que seguir simulando tiradas

4.16 rutina_tirada.h File Reference

Fichero que contiene la cabecera de la rutina que sigue cada proceso tirada.

This graph shows which files directly or indirectly include this file:



Functions

- void [proc_tirada](#) (int id, int *pipe)
Rutina del proceso monitor.

4.16.1 Detailed Description

Fichero que contiene la cabecera de la rutina que sigue cada proceso tirada.

Este fichero contiene la cabecera de la rutina que sigue cada proceso de tiradas

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.16.2 Function Documentation

4.16.2.1 void [proc_tirada](#) (int *id*, int * *pipe*)

Rutina del proceso monitor.

Funcion que ejecuta el proceso monitor nada más haberse bifurcado

Parameters

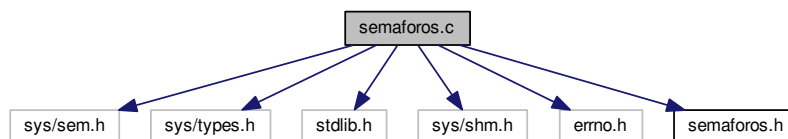
<i>id</i>	Id del caballo que realiza las tiradas
<i>pipe</i>	Tubería por la que se comunica el tipo de tirada al caballo.

4.17 semaforos.c File Reference

Funciones personales acerca de semaforos. Implementacion.

```
#include <sys/sem.h>
#include <sys/types.h>
#include <stdlib.h>
#include <sys/shm.h>
#include <errno.h>
#include "semaforos.h"
```

Include dependency graph for semaforos.c:



Functions

- int [inicializar_semaforo](#) (int semid, unsigned short *array)
Inicializa los semaforos indicados.
- int [borrar_semaforo](#) (int semid)
Borra un semáforo.
- int [crear_semaforo](#) (key_t key, int size, int *semid)
Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.
- int [down_semaforo](#) (int semid, int num_sem, int undo)
Baja el semaforo indicado.
- int [down_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Baja todos los semaforos del array indicado por active.
- int [up_semaforo](#) (int semid, int num_sem, int undo)
Sube el semaforo indicado.
- int [up_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Sube todos los semaforos del array indicado por active.

4.17.1 Detailed Description

Funciones personales acerca de semaforos. Implementacion.

Este fichero contiene el código fuente de las funciones en la interfaz [semaforos.h](#).

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

4.17.2 Function Documentation

4.17.2.1 `int borrar_semaforo (int semid)`

Borra un semáforo.

Parameters

<i>semid</i>	Identificador del semaforo.
--------------	-----------------------------

Returns

OK si todo fue correcto, ERROR en caso de error.

4.17.2.2 `int crear_semaforo (key_t key, int size, int * semid)`

Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.

Parameters

<i>key</i>	Clave precompartida del semaforo.
<i>size</i>	Tamaño del semaforo.
<i>semid</i>	Identificador del semaforo.

Returns

ERROR en caso de error, 0 si ha creado el semaforo, 1 si ya estaba creado.

4.17.2.3 int down_multiple_semaforo (int *semid*, int *size*, int *undo*, int * *active*)

Baja todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

int: OK si todo fue correcto, ERROR en caso de error.

4.17.2.4 int down_semaforo (int *semid*, int *num_sem*, int *undo*)

Baja el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

Returns

OK si todo fue correcto, ERROR en caso de error.

4.17.2.5 int inicializar_semaforo (int *semid*, unsigned short * *array*)

Inicializa los semaforos indicados.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>array</i>	Valores iniciales.

Returns

OK si todo fue correcto, ERROR en caso de error.

4.17.2.6 int up_multiple_semaforo (int *semid*, int *size*, int *undo*, int * *active*)

Sube todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

int: OK si todo fue correcto, ERROR en caso de error.

4.17.2.7 int up_semaforo (int *semid*, int *num_sem*, int *undo*)

Sube el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

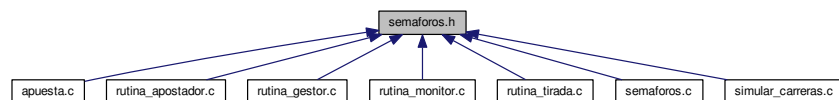
Returns

OK si todo fue correcto, ERROR en caso de error.

4.18 semaforos.h File Reference

Interfaz de funciones personales acerca de semaforos.

This graph shows which files directly or indirectly include this file:



Macros

- #define OK EXIT_SUCCESS
- #define ERROR -1

Functions

- int [inicializar_semaforo](#) (int semid, unsigned short *array)
Inicializa los semaforos indicados.
- int [borrar_semaforo](#) (int semid)
Borra un semáforo.
- int [crear_semaforo](#) (key_t key, int size, int *semid)
Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.
- int [down_semaforo](#) (int semid, int num_sem, int undo)
Baja el semaforo indicado.
- int [down_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Baja todos los semaforos del array indicado por active.
- int [up_semaforo](#) (int semid, int num_sem, int undo)
Sube el semaforo indicado.
- int [up_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Sube todos los semaforos del array indicado por active.

4.18.1 Detailed Description

Interfaz de funciones personales acerca de semaforos.

Este fichero contiene las funciones personales acerca de semaforos.

Author

Profesores de SOPER (EDIT: Rafael Sánchez & Sergio Galán).

Version

1.0

Date

06-04-2018

4.18.2 Macro Definition Documentation

4.18.2.1 #define ERROR -1

Error

4.18.2.2 #define OK EXIT_SUCCESS

Todo fue bien

4.18.3 Function Documentation

4.18.3.1 int borrar_semaforo (int semid)

Borra un semáforo.

Parameters

<i>semid</i>	Identificador del semaforo.
--------------	-----------------------------

Returns

OK si todo fue correcto, ERROR en caso de error.

4.18.3.2 int crear_semaforo (key_t key, int size, int * semid)

Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.

Parameters

<i>key</i>	Clave precompartida del semaforo.
<i>size</i>	Tamaño del semaforo.
<i>semid</i>	Identificador del semaforo.

Returns

ERROR en caso de error, 0 si ha creado el semaforo, 1 si ya estaba creado.

4.18.3.3 int down_multiple_semaforo (int semid, int size, int undo, int * active)

Baja todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

int: OK si todo fue correcto, ERROR en caso de error.

4.18.3.4 int down_semaforo (int semid, int num_sem, int undo)

Baja el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

Returns

OK si todo fue correcto, ERROR en caso de error.

4.18.3.5 int inicializar_semaforo (int *semid*, unsigned short * *array*)

Inicializa los semaforos indicados.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>array</i>	Valores iniciales.

Returns

OK si todo fue correcto, ERROR en caso de error.

4.18.3.6 int up_multiple_semaforo (int *semid*, int *size*, int *undo*, int * *active*)

Sube todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

int: OK si todo fue correcto, ERROR en caso de error.

4.18.3.7 int up_semaforo (int *semid*, int *num_sem*, int *undo*)

Sube el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

Returns

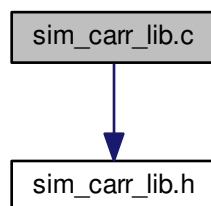
OK si todo fue correcto, ERROR en caso de error.

4.19 sim_carr_lib.c File Reference

Definición de las variables externas de `sim_carr_lib.h`.

```
#include "sim_carr_lib.h"
```

Include dependency graph for `sim_carr_lib.c`:



Variables

- int `num_proc`
- int `n_apos`
- int `n_cab`
- int `n_vent`
- double `din`

4.19.1 Detailed Description

Definición de las variables externas de `sim_carr_lib.h`.

Este fichero contiene la definición de las variables externas de [sim_carr_lib.h](#)

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.19.2 Variable Documentation

4.19.2.1 double din

Dinero inicial de los apostadores la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.19.2.2 int n_apos

Numero de apostadores en la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.19.2.3 int n_cab

Numero de caballos en la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.19.2.4 int n_vent

Numero de ventanillas en la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

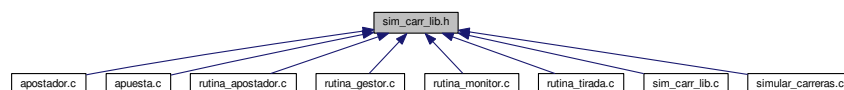
4.19.2.5 int num_proc

Numero de procesos hijo: $n_cab + n_apos + 2$ (Gestor y monitor). Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.20 sim_carr_lib.h File Reference

Macros, variables globales y estructuras generales del proyecto.

This graph shows which files directly or indirectly include this file:



Classes

- struct [msgtir](#)
Estructura del mensaje de tirada.
- struct [msgapues](#)
Informacion del mensaje de apuesta.

Macros

- #define [SIGSTART](#) SIGUSR1
- #define [SIGTHROW](#) SIGUSR2
- #define [KEY_APUES_Q](#) 1300
- #define [KEY_TIR_Q](#) 1310
- #define [KEY_CAB_SHM](#) 1400
- #define [KEY_APOS_SHM](#) 1410
- #define [KEY_MON_SEM](#) 1500
- #define [KEY_TUR_SEM](#) 1510
- #define [KEY_CAB_SEM](#) 1520
- #define [KEY_GEN_SEM](#) 1530
- #define [READ](#) 0
- #define [WRITE](#) 1
- #define [NORMAL](#) 0
- #define [GANADORA](#) 1
- #define [REMONTAR](#) 2
- #define [PATH](#) "/bin/bash"
- #define [MAX_NAME](#) 20
- #define [RUTA_FICHERO_APUESTAS](#) "text/apuestas.txt"
- #define [TIEMPO_PRE_CARR](#) 30
- #define [MAX_APOS](#) 100

Variables

- int [n_cab](#)
- int [n_apos](#)
- int [n_vent](#)
- double [din](#)
- int [num_proc](#)

4.20.1 Detailed Description

Macros, variables globales y estructuras generales del proyecto.

Este fichero contiene las macros, variables globales y estructuras necesarias para el proyecto.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

09-05-2018

4.20.2 Macro Definition Documentation

4.20.2.1 `#define GANADORA 1`

Tirada ganadora del caballo

4.20.2.2 `#define KEY_APOS_SHM 1410`

Clave para generar la memoria compartida de los apostadores

4.20.2.3 `#define KEY_APUES_Q 1300`

Clave para generar la cola de apuestas

4.20.2.4 `#define KEY_CAB_SEM 1520`

Clave para generar el semaforo que controla el acceso a la memoria compartida de los caballos

4.20.2.5 `#define KEY_CAB_SHM 1400`

Clave para generar la memoria compartida de los caballos

4.20.2.6 `#define KEY_GEN_SEM 1530`

Clave para generar el semaforo general que hace esperar a cada proceso al comienzo hasta que el principal le da paso

4.20.2.7 `#define KEY_MON_SEM 1500`

Clave para generar el semaforo del monitor

4.20.2.8 `#define KEY_TIR_Q 1310`

Clave para generar la cola de tiradas de los caballos

4.20.2.9 `#define KEY_TUR_SEM 1510`

Clave para generar el semaforo que controla el turno de tirada

4.20.2.10 `#define MAX_APOS 100`

Numero maximo de apostadores

4.20.2.11 #define MAX_NAME 20

Tamaño maximo del nombre del apostador en el mensaje

4.20.2.12 #define NORMAL 0

Tirada normal del caballo

4.20.2.13 #define PATH "/bin/bash"

Path para generar las claves de distintos IPCS

4.20.2.14 #define READ 0

Opcion de lectura en la pipe

4.20.2.15 #define REMONTAR 2

Tirada remontadora del caballo

4.20.2.16 #define RUTA_FICHERO_APUESTAS "text/apuestas.txt"

Ruta del fichero donde se guardan las apuestas

4.20.2.17 #define SIGSTART SIGUSR1

Redefinicion de la señal SIGUSR1 que indica el comienzo de la carrea

4.20.2.18 #define SIGTHROW SIGUSR2

Redefinicion de la señal SIGUSR2 que hace que los hijos relaicen una tirada

4.20.2.19 #define TIEMPO_PRE_CARR 30

Tiempo pre-carrera para apostar

4.20.2.20 #define WRITE 1

Opcion de escritura en la pipe

4.20.3 Variable Documentation

4.20.3.1 double din

Dinero inicial de los apostadores la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.20.3.2 int n_apos

Numero de apostadores en la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.20.3.3 int n_cab

Numero de caballos en la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.20.3.4 int n_vent

Numero de ventanillas en la simulacion. Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.20.3.5 int num_proc

Numero de procesos hijo: $n_cab + n_apos + 2$ (Gestor y monitor). Se define en [sim_carr_lib.c](#) y se inicializa una vez validados los parametros de entrada

4.21 `simular_carreras.c` File Reference

Rutina que simula la carrera.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <sys/shm.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <stdbool.h>
#include <signal.h>
#include <unistd.h>
#include "sim_carr_lib.h"
#include "mylib.h"
#include "semaforos.h"
#include "caballo.h"
#include "apostador.h"
#include "apuesta.h"
#include "rutina_monitor.h"
#include "rutina_gestor.h"
#include "rutina_tirada.h"
#include "rutina_apostador.h"
```


4.21.2 Macro Definition Documentation

4.21.2.1 #define NUM_ARGS 6

Numero de argumentos de entrada

4.21.3 Function Documentation

4.21.3.1 void _killall (int *sig*)

Función que envía una señal a todos los hijos.

Envía la señal elegida a todos sus hijos

Parameters

<i>sig</i>	Señal que queremos mandar a todos los hijos
------------	---

4.21.3.2 void _sim_handler (int *sig*)

Manejador de señales de la simulación de la carrera.

Establece las acciones que deberá ejecutar el proceso cuando reciba ciertas señales

Parameters

<i>sig</i>	Señal recibida
------------	----------------

4.21.3.3 int main (int *argc*, char * *argv*[])

main

Programa principal del simulador de carreras.

Parameters

<i>argc</i>	Número de argumentos de entrada
<i>argv</i>	Argumentos de entrada

Returns

Valor de salida del programa

4.21.3.4 `void usage ()`

Función que muestra el uso del programa y los parámetros de entrada.

Imprime por pantalla el uso y los parámetros de entrada de la función

4.21.4 Variable Documentation

4.21.4.1 `Apostador*` `apostadores`

Puntero al array de apostadores

4.21.4.2 `Caballo*` `caballos`

Puntero al array de caballos

4.21.4.3 `pid_t` `gestor`

Id de proceso del gestor de apuestas

4.21.4.4 `pid_t` `monitor`

Id de proceso del monitor

4.21.4.5 `volatile bool` `running_principal = true`

Bandera que indica si el proceso sigue ejecutando la lógica de tiradas

Index

- [_Apostador, 5](#)
 - [ben, 5](#)
 - [din_rest, 5](#)
 - [nombre, 5](#)
 - [pid, 6](#)
 - [total, 6](#)
 - [_Apuesta, 6](#)
 - [apos, 7](#)
 - [c, 7](#)
 - [cab_id, 7](#)
 - [cantidad, 7](#)
 - [ventanilla, 7](#)
 - [_Caballo, 7](#)
 - [cotizacion, 8](#)
 - [dinero_apostado, 8](#)
 - [id, 8](#)
 - [last_tir, 8](#)
 - [pid, 8](#)
 - [posicion, 8](#)
 - [_apos_handler](#)
 - [rutina_apostador.c, 48](#)
 - [_apuesta_print](#)
 - [apuesta.c, 25](#)
 - [_gestor_handler](#)
 - [rutina_gestor.c, 51](#)
 - [_killall](#)
 - [simular_carreras.c, 74](#)
 - [_monitor_carrera](#)
 - [rutina_monitor.c, 54](#)
 - [_monitor_fin](#)
 - [rutina_monitor.c, 54](#)
 - [_monitor_handler](#)
 - [rutina_monitor.c, 54](#)
 - [_monitor_post_carrera](#)
 - [rutina_monitor.c, 55](#)
 - [_monitor_pre_carrera](#)
 - [rutina_monitor.c, 55](#)
 - [_rutina_ventanilla](#)
 - [rutina_gestor.c, 51](#)
 - [_sim_handler](#)
 - [simular_carreras.c, 74](#)
 - [_tirada_handler](#)
 - [rutina_tirada.c, 58](#)
 - [_ventanilla, 9](#)
 - [apostadores, 9](#)
 - [caballos, 9](#)
 - [msgqid, 9](#)
 - [ventanilla, 9](#)
- [APOS_ERROR](#)

- [apostador.h, 19](#)
- [apos](#)
 - [_Apuesta, 7](#)
- [apos_cmp_ben](#)
 - [apostador.c, 14](#)
 - [apostador.h, 20](#)
- [apos_destroy](#)
 - [apostador.c, 14](#)
 - [apostador.h, 20](#)
- [apos_get_ben](#)
 - [apostador.c, 15](#)
 - [apostador.h, 20](#)
- [apos_get_name](#)
 - [apostador.c, 15](#)
 - [apostador.h, 21](#)
- [apos_get_pid](#)
 - [apostador.c, 15](#)
 - [apostador.h, 21](#)
- [apos_get_total](#)
 - [apostador.c, 16](#)
 - [apostador.h, 21](#)
- [apos_incr_din_rest](#)
 - [apostador.c, 16](#)
 - [apostador.h, 22](#)
- [apos_init](#)
 - [apostador.c, 16](#)
 - [apostador.h, 22](#)
- [apos_new](#)
 - [apostador.c, 17](#)
 - [apostador.h, 22](#)
- [apos_refresh_total](#)
 - [apostador.c, 17](#)
 - [apostador.h, 22](#)
- [apos_set_ben](#)
 - [apostador.c, 17](#)
 - [apostador.h, 23](#)
- [apos_set_pid](#)
 - [apostador.c, 17](#)
 - [apostador.h, 23](#)
- [Apostador](#)
 - [apostador.h, 20](#)
- [apostador.c, 13](#)
 - [apos_cmp_ben, 14](#)
 - [apos_destroy, 14](#)
 - [apos_get_ben, 15](#)
 - [apos_get_name, 15](#)
 - [apos_get_pid, 15](#)
 - [apos_get_total, 16](#)
 - [apos_incr_din_rest, 16](#)

- apos_init, 16
 - apos_new, 17
 - apos_refresh_total, 17
 - apos_set_ben, 17
 - apos_set_pid, 17
- apostador.h, 18
 - APOS_ERROR, 19
 - apos_cmp_ben, 20
 - apos_destroy, 20
 - apos_get_ben, 20
 - apos_get_name, 21
 - apos_get_pid, 21
 - apos_get_total, 21
 - apos_incr_din_rest, 22
 - apos_init, 22
 - apos_new, 22
 - apos_refresh_total, 22
 - apos_set_ben, 23
 - apos_set_pid, 23
 - Apostador, 20
 - MAX_APOS_NAME, 19
 - MAX_CAB, 20
- apostadores
 - _ventanilla, 9
 - rutina_apostador.c, 49
 - simular_carreras.c, 75
- Apuesta
 - apuesta.h, 29
- apuesta.c, 24
 - _apuesta_print, 25
 - apuesta_destroy, 25
 - apuesta_execute, 25
 - apuesta_init, 27
 - apuesta_new, 27
 - apuesta_total, 27
- apuesta.h, 28
 - Apuesta, 29
 - apuesta_destroy, 29
 - apuesta_execute, 29
 - apuesta_init, 29
 - apuesta_new, 30
- apuesta_destroy
 - apuesta.c, 25
 - apuesta.h, 29
- apuesta_execute
 - apuesta.c, 25
 - apuesta.h, 29
- apuesta_init
 - apuesta.c, 27
 - apuesta.h, 29
- apuesta_new
 - apuesta.c, 27
 - apuesta.h, 30
- apuesta_total
 - apuesta.c, 27
- aredigits
 - mylib.c, 44
 - mylib.h, 46
- ben
 - _Apostador, 5
- borrar_semaforo
 - semaforos.c, 61
 - semaforos.h, 64
- c
 - _Apuesta, 7
- CAB_ERROR
 - caballo.h, 38
- cab_destroy
 - caballo.c, 31
 - caballo.h, 38
- cab_get_apostado
 - caballo.c, 31
 - caballo.h, 39
- cab_get_cot
 - caballo.c, 32
 - caballo.h, 39
- cab_get_id
 - caballo.c, 32
 - caballo.h, 39
- cab_get_last_tir
 - caballo.c, 32
 - caballo.h, 39
- cab_get_pid
 - caballo.c, 33
 - caballo.h, 40
- cab_get_pos
 - caballo.c, 33
 - caballo.h, 40
- cab_id
 - _Apuesta, 7
- cab_incr_apostado
 - caballo.c, 33
 - caballo.h, 40
- cab_new
 - caballo.c, 34
 - caballo.h, 41
- cab_set_cot
 - caballo.c, 34
 - caballo.h, 41
- cab_set_id
 - caballo.c, 34
 - caballo.h, 41
- cab_set_last_tir
 - caballo.c, 34
 - caballo.h, 42
- cab_set_pid
 - caballo.c, 36
 - caballo.h, 42
- cab_set_pos
 - caballo.c, 36
 - caballo.h, 42
- Caballo
 - caballo.h, 38
- caballo
 - msgapues, 10
- caballo.c, 30

- cab_destroy, 31
- cab_get_apostado, 31
- cab_get_cot, 32
- cab_get_id, 32
- cab_get_last_tir, 32
- cab_get_pid, 33
- cab_get_pos, 33
- cab_incr_apostado, 33
- cab_new, 34
- cab_set_cot, 34
- cab_set_id, 34
- cab_set_last_tir, 34
- cab_set_pid, 36
- cab_set_pos, 36
- caballo.h, 36
 - CAB_ERROR, 38
 - cab_destroy, 38
 - cab_get_apostado, 39
 - cab_get_cot, 39
 - cab_get_id, 39
 - cab_get_last_tir, 39
 - cab_get_pid, 40
 - cab_get_pos, 40
 - cab_incr_apostado, 40
 - cab_new, 41
 - cab_set_cot, 41
 - cab_set_id, 41
 - cab_set_last_tir, 42
 - cab_set_pid, 42
 - cab_set_pos, 42
 - Caballo, 38
- caballos
 - _ventanilla, 9
 - simular_carreras.c, 75
- cantidad
 - _Apuesta, 7
 - msgapues, 10
- cotizacion
 - _Caballo, 8
- crear_semaforo
 - semaforos.c, 61
 - semaforos.h, 65
- din
 - sim_carr_lib.c, 68
 - sim_carr_lib.h, 72
- din_rest
 - _Apostador, 5
- dinero_apostado
 - _Caballo, 8
- down_multiple_semaforo
 - semaforos.c, 61
 - semaforos.h, 65
- down_semaforo
 - semaforos.c, 62
 - semaforos.h, 65
- ERROR
 - semaforos.h, 64
- end
 - rutina_gestor.c, 52
- fin_carr
 - rutina_monitor.c, 55
- fin_pre_carr
 - rutina_monitor.c, 55
- GANADORA
 - sim_carr_lib.h, 70
- gestor
 - simular_carreras.c, 75
- id
 - _Caballo, 8
- inicializar_semaforo
 - semaforos.c, 62
 - semaforos.h, 66
- isfloat
 - mylib.c, 44
 - mylib.h, 46
- KEY_APOS_SHM
 - sim_carr_lib.h, 70
- KEY_APUES_Q
 - sim_carr_lib.h, 70
- KEY_CAB_SEM
 - sim_carr_lib.h, 70
- KEY_CAB_SHM
 - sim_carr_lib.h, 70
- KEY_GEN_SEM
 - sim_carr_lib.h, 70
- KEY_MON_SEM
 - sim_carr_lib.h, 70
- KEY_TIR_Q
 - sim_carr_lib.h, 70
- KEY_TUR_SEM
 - sim_carr_lib.h, 70
- last_tir
 - _Caballo, 8
- MAX_APOS_NAME
 - apostador.h, 19
- MAX_APOS_PRINT
 - rutina_monitor.c, 54
- MAX_APOS
 - sim_carr_lib.h, 70
- MAX_CAB
 - apostador.h, 20
- MAX_NAME
 - sim_carr_lib.h, 70
- main
 - simular_carreras.c, 74
- monitor
 - simular_carreras.c, 75
- msgapues, 10
 - caballo, 10
 - cantidad, 10
 - mtype, 10

- nombre, 10
- msgqid
 - _ventanilla, 9
- msgtir, 11
 - mtype, 11
 - tirada, 11
- mtype
 - msgapues, 10
 - msgtir, 11
- mylib.c, 43
 - aredigits, 44
 - isfloat, 44
 - randNum, 44
 - sigaddset_var, 44
- mylib.h, 45
 - aredigits, 46
 - isfloat, 46
 - randNum, 46
 - sigaddset_var, 47
- n_apos
 - sim_carr_lib.c, 68
 - sim_carr_lib.h, 72
- n_cab
 - sim_carr_lib.c, 68
 - sim_carr_lib.h, 72
- n_vent
 - sim_carr_lib.c, 68
 - sim_carr_lib.h, 72
- NORMAL
 - sim_carr_lib.h, 71
- NUM_ARGS
 - simular_carreras.c, 74
- nombre
 - _Apostador, 5
 - msgapues, 10
- num_proc
 - sim_carr_lib.c, 68
 - sim_carr_lib.h, 72
- OK
 - semaforos.h, 64
- PATH
 - sim_carr_lib.h, 71
- pid
 - _Apostador, 6
 - _Caballo, 8
- posicion
 - _Caballo, 8
- proc_apostador
 - rutina_apostador.c, 48
 - rutina_apostador.h, 50
- proc_gestor
 - rutina_gestor.c, 51
 - rutina_gestor.h, 53
- proc_monitor
 - rutina_monitor.c, 55
 - rutina_monitor.h, 57
- proc_tirada
 - rutina_tirada.c, 58
 - rutina_tirada.h, 59
- READ
 - sim_carr_lib.h, 71
- REMONTAR
 - sim_carr_lib.h, 71
- ruta_FICHERO_APUESTAS
 - sim_carr_lib.h, 71
- randNum
 - mylib.c, 44
 - mylib.h, 46
- running_apostador
 - rutina_apostador.c, 49
- running_principal
 - simular_carreras.c, 75
- running_tirada
 - rutina_tirada.c, 58
- rutina_apostador.c, 47
 - _apos_handler, 48
 - apostadores, 49
 - proc_apostador, 48
 - running_apostador, 49
- rutina_apostador.h, 49
 - proc_apostador, 50
- rutina_gestor.c, 50
 - _gestor_handler, 51
 - _rutina_ventanilla, 51
 - end, 52
 - proc_gestor, 51
- rutina_gestor.h, 52
 - proc_gestor, 53
- rutina_monitor.c, 53
 - _monitor_carrera, 54
 - _monitor_fin, 54
 - _monitor_handler, 54
 - _monitor_post_carrera, 55
 - _monitor_pre_carrera, 55
 - fin_carr, 55
 - fin_pre_carr, 55
 - MAX_APOS_PRINT, 54
 - proc_monitor, 55
- rutina_monitor.h, 56
 - proc_monitor, 57
- rutina_tirada.c, 57
 - _tirada_handler, 58
 - proc_tirada, 58
 - running_tirada, 58
- rutina_tirada.h, 59
 - proc_tirada, 59
- SIGSTART
 - sim_carr_lib.h, 71
- SIGTHROW
 - sim_carr_lib.h, 71
- semaforos.c, 60
 - borrar_semaforo, 61
 - crear_semaforo, 61

- down_multiple_semaforo, 61
- down_semaforo, 62
- inicializar_semaforo, 62
- up_multiple_semaforo, 62
- up_semaforo, 63
- semaforos.h, 63
 - borrar_semaforo, 64
 - crear_semaforo, 65
 - down_multiple_semaforo, 65
 - down_semaforo, 65
 - ERROR, 64
 - inicializar_semaforo, 66
 - OK, 64
 - up_multiple_semaforo, 66
 - up_semaforo, 66
- sigaddset_var
 - mylib.c, 44
 - mylib.h, 47
- sim_carr_lib.c, 67
 - din, 68
 - n_apos, 68
 - n_cab, 68
 - n_vent, 68
 - num_proc, 68
- sim_carr_lib.h, 68
 - din, 72
 - GANADORA, 70
 - KEY_APOS_SHM, 70
 - KEY_APUES_Q, 70
 - KEY_CAB_SEM, 70
 - KEY_CAB_SHM, 70
 - KEY_GEN_SEM, 70
 - KEY_MON_SEM, 70
 - KEY_TIR_Q, 70
 - KEY_TUR_SEM, 70
 - MAX_APOS, 70
 - MAX_NAME, 70
 - n_apos, 72
 - n_cab, 72
 - n_vent, 72
 - NORMAL, 71
 - num_proc, 72
 - PATH, 71
 - READ, 71
 - REMONTAR, 71
 - RUTA_FICHERO_APUESTAS, 71
 - SIGSTART, 71
 - SIGTHROW, 71
 - TIEMPO_PRE_CARR, 71
 - WRITE, 71
- simular_carreras.c, 72
 - _killall, 74
 - _sim_handler, 74
 - apostadores, 75
 - caballos, 75
 - gestor, 75
 - main, 74
 - monitor, 75
 - NUM_ARGS, 74
 - running_principal, 75
 - usage, 74
- TIEMPO_PRE_CARR
 - sim_carr_lib.h, 71
- tirada
 - msgtir, 11
- total
 - _Apostador, 6
- up_multiple_semaforo
 - semaforos.c, 62
 - semaforos.h, 66
- up_semaforo
 - semaforos.c, 63
 - semaforos.h, 66
- usage
 - simular_carreras.c, 74
- ventanilla
 - _Apuesta, 7
 - _ventanilla, 9
- WRITE
 - sim_carr_lib.h, 71