

# Swimming



# Manager

## GRUPO 6+2

JUAN LUIS ÁLVAREZ HERRADÓN

LARA CARRIÓN OVEJERO

CHARLES FLORES ESPINOZA

CLAUDIA GIL NAVARRO

VÍKTOR SHAMEL JACYNICZ GARCÍA

JAIME MAYORDOMO MORENO

MARTA OLIVER VALIENTE

ÁNGEL LUIS ORTIZ FOLGADO

---

# CONTENIDO

---

**1. INTRODUCCIÓN .....3**

**2. REQUISITOS .....5**

**3. ESPECIFICACIÓN DEL PROYECTO .....8**

**4. TABLAS DE RIESGOS.....15**

**5. PLANIFICACIÓN.....18**

---

# 1. INTRODUCCIÓN

---

---

## ACERCA DE SWIMMING MANAGER

---

*Swimming Manager* es un proyecto que nace con la finalidad de desarrollar un gestor de deportistas enfocado a competiciones de natación.

Éste está destinado principalmente a aquellos que se dedican a llevar un seguimiento de deportistas en pruebas de natación, tales como directores deportivos, entrenadores personales, clubes, federaciones y/o árbitros de natación.

*Swimming Manager* ofrece una gran variedad de funciones, entre las que cabe destacar las siguientes:

- ✓ Crear bases de datos de nadadores
- ✓ Añadir información relevante sobre los deportistas, como son datos personales, marcas, club al que pertenece, etc.
- ✓ Clasificaciones por estilos, edad y marcas
- ✓ Comparación de marcas entre deportistas
- ✓ Listas de patrocinadores por clubes

En cuanto a objetivos se refiere, *Swimming Manager* se enfoca en varios de ellos, detallados en el siguiente punto.

### OBJETIVOS

Los objetivos de este gestor de nadadores pueden resumirse en:

- Facilitar la creación de una base de datos ordenada con los nombres y datos personales de todos los nadadores pertenecientes a un club o federación.
- Acceder fácilmente a una clasificación ordenada de los nadadores por pruebas y estilos.
- Facilitar la obtención de información referente a un deportista, club deportivo u organización.
- Agilizar trámites en lo referente a pruebas de un día o competiciones de alto nivel, así como obtener información detallada de las mismas.

## Acerca del proyecto

---

Este proyecto ha sido desarrollado en el contexto de la asignatura de Ingeniería del Software de 2º Curso del Grado en Ingeniería Informática de la Universidad Complutense de Madrid, bajo la supervisión del profesor Pablo Gervás.

El objetivo principal de nuestro proyecto es unir técnicas, recursos y personal en un solo entorno para conseguir resultados rápidos, ordenados y eficaces, enfocado principalmente a profesionales que trabajan en un escenario deportivo especializado en pruebas de natación.

Éste, al estar destinado principalmente a personas que tienen que llevar un seguimiento diario de las competiciones de natación y que suponemos van a dar un uso continuado e intensivo al programa, la interfaz está diseñada de forma sencilla e intuitiva, con el fin de facilitar el manejo del mismo y el seguimiento de deportistas al usuario.

El modelo de proceso de desarrollo utilizado será un desarrollo basado en componentes, reutilizando software para simplificar las pruebas y el mantenimiento, y de esta forma consiguiendo una mayor calidad al finalizar el proyecto.

El lenguaje utilizado para diseñar el programa será Java, porque nos resulta más apropiado debido a su orientación a objetos y la facilidad de uso de las interfaces, además de ser un lenguaje muy utilizado que ofrece gran cantidad de librerías.

De esta forma, utilizaremos dos entornos de programación:

- Eclipse, porque es el entorno que más ayudas ofrece al escribir código y es el que nos resulta más conocido.
- NetBeans, por la comodidad de su uso para diseñar interfaces gráficas de usuario, es ideal para ahorrar tiempo sustituyendo a Eclipse en esta labor.

---

## 2. REQUISITOS

---

*1. Requisitos no funcionales - ¿Cómo lo hace el sistema?*

*2. Requisitos funcionales - ¿Qué hace el sistema?: Casos de uso.*

---

### REQUISITOS NO FUNCIONALES

---

#### RENDIMIENTO DEL SISTEMA

El sistema ha de cumplir unos requisitos no funcionales esenciales para su correcto funcionamiento:

- Debe visualizarse, ser compatible y funcionar correctamente en cualquier sistema operativo y versión, especialmente con Windows, Mac y Linux.
- Debe ejecutarse nada más abrirlo en un tiempo no superior a 5 segundos, no tardando más de 30 segundos en empezar a trabajar tras la carga de información almacenada.
- Debe realizar una búsqueda en un tiempo inferior a 5 segundos.
- Al dar de alta un nuevo deportista, el tiempo de espera no debe ser mayor de 3 minutos, ya que en este caso el sistema lanzará un mensaje de aviso pidiendo todos los datos nuevamente.
- Al dar de baja un deportista, el sistema no debe tardar más de 5 segundos en eliminar toda la información acerca de éste.
- Si ocurre un fallo en el sistema, el tiempo de restauración máximo aceptable será de 3 minutos hasta volver al punto en el que se encontraba el usuario antes de dicho fallo.

## SEGURIDAD

En esta versión del proyecto, para poder acceder al sistema no es necesario introducir un nombre de usuario y contraseña, por lo que cualquier usuario podría tener acceso a datos y modificaciones. En versiones futuras, el sistema se comportará de la siguiente manera:

- Para poder acceder al sistema y a la información almacenada será necesario introducir un nombre de usuario y contraseña correctos que posteriormente serán verificados por el sistema.
- Cualquiera de los administradores podrá también acceder al sistema con su propio nombre de usuario y contraseña.

## FIABILIDAD

El sistema debe ser fiable, ya que de otro modo puede ocasionar fallos que conlleven problemas difíciles de solucionar una vez desarrollado el software.

Los fallos de funcionamiento de un sistema pueden tener su origen en errores producidos al diseñar el software o problemas con el hardware.

Para aumentar la fiabilidad del sistema es conveniente prevenir los fallos y hacer que el sistema sea tolerante a ellos, siguiendo éste en funcionamiento aunque se produzcan errores, con el fin de evitar problemas tanto al usuario como a los desarrolladores.

## MANTENIBILIDAD

La mantenibilidad es la facilidad con la que un sistema o componente software puede ser modificado para corregir fallos, mejorar su funcionamiento u otros atributos o adaptarse a cambios en el entorno.

El tipo de mantenimiento del proyecto hemos estimado que será mensual, previendo posibles fallos y reparando los ya existentes detectados en anteriores iteraciones.

También realizaremos las modificaciones oportunas si algún error es detectado por el usuario y lanzaremos una nueva versión previamente revisada.

## PORTABILIDAD

Por su amplia compatibilidad con la mayoría de los sistemas operativos actuales y nuestro conocimiento acerca del mismo, hemos elegido diseñar nuestro sistema empleando el lenguaje de programación Java.

Para su desarrollo, utilizaremos Eclipse y NetBeans como entornos de programación; el primero, por su facilidad de uso y familiaridad con el entorno, y el segundo, por la comodidad que ofrece en la creación de interfaces gráficas.

---

## 3. ESPECIFICACIÓN DEL PROYECTO

---

---

### ACTOR PRINCIPAL

---

En nuestro caso, el actor principal es un usuario que se encargue de llevar el seguimiento de los deportistas dados de alta en el programa. Sus objetivos serán:

- ✓ Dar de alta a un nadador<sup>1</sup>.
- ✓ Dar de baja a un nadador.
- ✓ Añadir marca de un nadador<sup>2</sup>.
- ✓ Eliminar marca de un nadador.
- ✓ Ver la información asociada a un nadador.
- ✓ Buscar nadadores introducidos en el programa por nombre, edad, sexo, nacionalidad, nº federativo, estilo, distancia, marca o récord.
- ✓ Comparar marcas entre nadadores.
- ✓ Guardar la información introducida en el programa en un archivo de texto.
- ✓ Cargar datos de un archivo de texto previamente generado con el programa.
- ✓ Mostrar una lista de todos los nadadores que hay actualmente dados de alta.

<sup>1</sup>Para dar de alta a un nuevo nadador, se solicitará al usuario la siguiente información: nombre, edad, sexo, nacionalidad y nº federativo.

<sup>2</sup>Las marcas se contabilizarán en minutos y segundos. Se almacenarán las 10 más recientes de cada nadador además de su mejor marca hasta la fecha.

---

### CASOS DE USO

---

- ❖ Dar de alta a un nadador
  - Objetivo: Introducir en la base de datos del programa a un nuevo nadador aportando su nombre, edad, sexo, nacionalidad y nº federativo.
  - Precondición: El programa ha sido iniciado correctamente.
  - Postcondición sin fallo: La información del nuevo nadador es añadida al programa.
  - Postcondición con fallo: El nadador ya había sido introducido previamente. Se informa del error.
  - Actores: Usuario.



- Secuencia normal:
  1. Se pulsa *Editar=>Dar de alta* en la barra de opciones del programa.
  2. Se muestra un formulario con los campos *Nombre, Edad, Sexo, Nacionalidad* y *Nº federativo* que el usuario debe completar.
  3. Una vez han sido rellenados todos los datos, se comprueba si ya existe algún nadador en el sistema con el mismo nombre. En caso afirmativo, se pasa a (S1). En caso contrario, se almacenan los datos.
- Secuencia alternativa:
  - S1. Se informa de que el nadador que el usuario intenta almacenar ya se encuentra dado de alta en el sistema.

❖ Dar de baja a un nadador

- Objetivo: Eliminar de la base de datos del programa a un nadador seleccionado por el usuario.
- Precondición: El programa ha sido iniciado correctamente.
- Postcondición sin fallo: La información del nadador se elimina del programa.
- Postcondición con fallo: No se ha introducido ningún nadador. Se informa del error.
- Actores: Usuario.
- Secuencia normal:
  1. Se pulsa *Editar=>Dar de baja* en la barra de opciones del programa.
  2. Se solicita al usuario que elija el nadador que desea dar de baja de una lista con los nombres de los que actualmente se encuentran almacenados. En caso de no haber ninguno dado de alta, se pasa a (S1).
  3. Una vez ha sido seleccionado el nadador, el programa lo elimina de la base de datos.
- Secuencia alternativa:
  - S1. Se informa de que no hay ningún nadador dado de alta en el programa.

❖ Añadir marca de un nadador

- Objetivo: Añadir una nueva marca para algún nadador ya dado de alta.
- Precondición: El usuario ha introducido algún nadador previamente en el programa.
- Postcondición sin fallo: Se añade la nueva marca al nadador seleccionado por el usuario.

- Postcondición con fallo: No se ha introducido ningún nadador en el programa. Se informa del error.
- Actores: Usuario.
- Secuencia normal:
  1. Se pulsa *Editar=>Nadador=>Añadir marca* en el menú de opciones del programa.
  2. Se solicita al usuario que elija al nadador de entre una lista con los nombres de los que actualmente se encuentran almacenados y rellene el campo *Marca* con el nuevo tiempo. En caso de no haber ninguno dado de alta, se pasa a (S1).
  3. Una vez ha sido seleccionado el nadador e introducida la nueva marca, se comprueba si ya se le habían asignado 10 distintas al deportista correspondiente. En caso afirmativo, se elimina la más antigua y se añade la actual. En caso contrario, simplemente se añade la nueva marca.
  4. Se compara la marca récord del nadador con la nueva que ha sido introducida. En caso de ser mejor que la que hay almacenada, ésta se actualiza con la nueva marca. En caso contrario, no se realiza ningún cambio en ese campo
- Secuencia alternativa:
  - S1. Se informa de que no hay ningún nadador dado de alta.

#### ❖ Eliminar marca de un nadador

- Objetivo: Eliminar una marca seleccionada de un nadador ya dado de alta.
- Precondición: El usuario ha introducido algún nadador previamente en el programa.
- Postcondición sin fallo: Se elimina la marca seleccionada por el usuario del nadador correspondiente.
- Posctondición con fallo: No se ha introducido ningún nadador en el programa. Se informa del error.
- Actores: Usuario.
- Secuencia normal:
  1. Se pulsa *Editar=>Nadador=>Eliminar marca* en el menú de opciones del programa.
  1. Se solicita al usuario que elija al nadador cuya marca desee eliminar de una lista con los nombres de los que actualmente se encuentran almacenados. En caso de no haber ningún nadador, se pasa a (S1). En caso de no haberse introducido ninguna marca para ese nadador, se pasa a (S2).
  2. Se muestra una lista de las marcas asignadas al nadador seleccionado y se pide al usuario que seleccione aquella que desea eliminar.
  3. Una vez el usuario ha seleccionado la marca deseada, el programa elimina dicha marca del nadador.

- Secuencias alternativas:
  - S1. Se informa de que no hay ningún nadador dado de alta.
  - S2. Se informa de que aún no se le ha asignado ninguna marca al nadador seleccionado.

❖ Ver la información asociada a un nadador

- Objetivo: Mostrar en pantalla la información existente de un nadador seleccionado por el usuario.
- Precondición: El usuario ha introducido algún nadador previamente en el programa.
- Postcondición sin fallo: Se muestra la información asociada al nadador que el usuario ha seleccionado.
- Postcondición con fallo: No se ha introducido ningún nadador en el programa. Se informa del error.
- Actores: Usuario.
- Secuencia normal:
  1. Se pulsa *Editar=>Nadador=>Información* en la barra de opciones del programa.
  2. Se solicita al usuario que elija al nadador cuya información desee visualizar de una lista con los nombres de los que actualmente se encuentran almacenados. En caso de no haber ningún nadador, se pasa a (S1).
  3. Una vez ha sido seleccionado el nadador, se muestran por pantalla sus datos, marcas introducidas y récord.
- Secuencia alternativa:
  - S1. Se informa de que no hay ningún nadador dado de alta.

❖ Buscar nadadores introducidos en el programa por nombre, edad, sexo, nacionalidad, nº federativo, estilo, distancia, marca o récord

- Objetivo: Buscar nadadores por medio del criterio seleccionado por el usuario (nombre, edad, sexo, nacionalidad, nº federativo, estilo, distancia, marca o récord).
- Precondición: El usuario ha introducido a algún nadador previamente en el programa.
- Postcondición sin fallo: Se muestra una lista con los nombres de los nadadores que el programa ha encontrado según el criterio que el usuario ha seleccionado.
- Postcondición con fallo: No se ha encontrado ningún nadador que cumpla el criterio elegido. Se informa al usuario de ello.
- Actores: Usuario.
- Secuencia normal:

1. Se pulsa *Buscar por...* y se selecciona la opción deseada en la barra de opciones del programa: *Nombre, Nacionalidad, Sexo, Estilo, Distancia, Nº federativo, Edad, Marca, Récord*.
  2. El usuario rellena el campo seleccionado.
  3. Una vez ha sido introducido el campo correspondiente, el programa busca entre los nadadores almacenados aquellos que se correspondan con el criterio de búsqueda establecido. Se muestra una lista con los nombres de los nadadores encontrados o un mensaje indicando que no existe ninguno que cumpla lo establecido. En caso de no haberse almacenado ningún nadador, se pasa a (S1).
- Secuencia alternativa:
    - S1. Se informa de que no hay ningún nadador dado de alta en el sistema.

❖ Comparar marcas entre nadadores

- Objetivo: Comparar la información de dos nadadores seleccionados por el usuario.
- Precondición: El usuario ha dado de alta a dos o más nadadores en el sistema.
- Postcondición sin fallo: Se muestra por pantalla los datos y marcas de los nadadores elegidos.
- Postcondición con fallo: No hay dos o más nadadores dados de alta y no puede realizarse una comparación. Se informa del error.
- Actores: Usuario.
- Secuencia normal:
  1. Se pulsa *Comparar* en la barra de opciones del programa.
  2. El usuario selecciona los dos nadadores que quiere comparar de una lista con los nombres de los que se encuentran actualmente dados de alta. En caso de no haber dos o más almacenados en el sistema, se pasa a (S1).
  3. Una vez seleccionados, el programa muestra por pantalla los datos, marcas y récords de ambos deportistas.
- Secuencia alternativa:
  - S1. Se informa de que no se han dado de alta a suficientes nadadores como para poder hacer una comparación.

❖ Guardar la información introducida en el programa en un archivo texto

- Objetivo: Almacenar la información que contenga el programa actualmente en un archivo de texto para su posterior uso en otra ocasión.

- Precondición: El usuario ha dado de alta a algún nadador en el sistema y ha modificado la información contenida desde la última vez que se procedió a guardar.
- Postcondición sin fallo: Se guarda la información contenida en el programa en un archivo de texto.
- Postcondición con fallo: No hay ningún nadador dado de alta en el sistema. Se informa del error.
- Actores: Usuario.
- Secuencia normal:
  1. Se pulsa *Archivo=>Guardar como* en la barra de opciones del programa.
  2. Si no se ha guardado la información en un fichero desde el inicio de sesión en el programa, se solicitará que se introduzca un nombre y una ubicación del archivo en el que se quiere almacenar los datos. Si no, se procederá a reescribir el fichero seleccionado previamente. En caso de no haber ningún nadador dado de alta en el sistema, se pasa a (S1).
  3. Los datos de los nadadores y sus correspondientes marcas y récords son almacenados en el archivo de texto.
- Secuencia alternativa:
  - S1. Se informa de que no hay ningún nadador dado de alta actualmente.

- ❖ Cargar datos de un archivo de texto generado previamente por el programa
  - Objetivo: Introducir datos en el programa por medio de un archivo de texto que ya habíamos generado en otra ocasión y así poder visualizarlo y/o modificarlo.
  - Precondición: Se ha guardado la información previamente en un archivo de texto mediante el programa.
  - Postcondición sin fallo: La información contenida en el archivo de texto se introduce en el programa
  - Postcondición con fallo: El archivo seleccionado no es válido. Se notifica al usuario del error.
  - Actores: Usuario.
  - Secuencia normal:
    1. Se pulsa *Archivo=>Abrir* en la barra de opciones del programa.
    2. El usuario selecciona el archivo de texto cuya información desea cargar en el programa.
    3. El programa lee el contenido del archivo de texto e intenta introducir la información en el sistema. En caso de producirse un error, se pasa a (S1). En caso contrario, se notifica que el archivo se ha cargado correctamente.
  - Secuencia alternativa:

S1. Se informa de que el archivo seleccionado no es válido y se pide al usuario que intente abrir uno que sí lo sea.

- ❖ Mostrar una lista de todos los nadadores que hay actualmente dados de alta
  - Objetivo: Mostrar en pantalla los datos, marcas y récords de todos los nadadores que hay dados de alta en el programa hasta el momento.
  - Precondición: Se ha dado de alta a algún nadador en el sistema,
  - Postcondición sin fallo: Se muestra una lista de los nadadores dados de alta con toda la información disponible.
  - Postcondición con fallo: No se ha dado de alta a ningún nadador. Se notifica al usuario del error.
  - Actores: Usuario.
- Secuencia normal:
  1. Se pulsa *Archivo=>Mostrar todo* en la barra de opciones del programa.
  2. Se intenta mostrar una lista con el nombre, edad, sexo, nacionalidad y nº federativo de cada nadador. En caso de no encontrarse ninguno dado de alta, se pasa a (S2). En caso contrario, se muestra la información.
- Secuencia alternativa:

S1. Se informa de que no hay ningún usuario dado de alta en el programa.

## 4. TABLAS DE RIESGOS

### TIPOS DE RIESGOS

Riesgo	Tipo de Riesgo	Descripción
Falta de trabajo en algún miembro del equipo.	Proyecto.	Alguno de los miembros del grupo no es capaz de completar las tareas que tiene asignadas.
Incompatibilidad de hardware.	Proyecto.	El programa puede estar diseñado para un hardware específico que no corresponde con el que el usuario dispone.
Cambio en los Requerimientos.	Proyecto, Producto.	Puede haber un cambio en los requerimientos del producto.
Retraso en el cumplimiento de los plazos.	Proyecto.	Es posible que no se puedan cumplir los plazos requeridos para entregar el proyecto a tiempo.
Compatibilidad con el sistema operativo	Proyecto, Producto.	Es posible que el usuario no utilice el sistema operativo para el que el programa está preparado.
Competencia.	Negocio.	Puede haber mejores programas y más eficientes en el mercado.
Coordinación.	Proyecto.	Falta de coordinación a la hora de repartir las labores de cada miembro del grupo.
Falta de conocimiento.	Proyecto.	El nivel de conocimiento de bases de datos es insuficiente para continuar el proyecto.
Librerías.	Proyecto.	No se encuentran librerías suficientes para elaborar un diseño eficiente del producto.
Almacenamiento de datos.	Proyecto.	El método de almacenar los datos es ineficiente o no funciona.
Introducción de datos.	Proyecto.	El método de introducción de datos es poco intuitivo y tiene mal diseño.
Utilidad del programa.	Proyecto, Producto, Negocio.	El colectivo al que va dirigido el programa no le encuentre utilidad.
Acceso a la información.	Proyecto, Producto.	Poder acceder y manipular la información no esta restringido a ciertos usuarios.

## PROBABILIDAD DE RIESGOS

Riesgo	Probabilidad	Severidad
Falta de trabajo en algún miembro del equipo.	Frecuente	Seria
Incompatibilidad de hardware.	Improbable	Menor
Cambio en los Requerimientos.	Remota	Seria
Retraso en el cumplimiento de los plazos.	Probable	Crítica
Compatibilidad con el sistema operativo	Improbable	Catastrófica
Competencia.	Ocasional	Menor
Coordinación.	Frecuente	Seria
Falta de conocimiento.	Probable	Seria
Librerías.	Ocasional	Seria
Almacenamiento de datos.	Probable	Critica
Introducción de datos.	Frecuente	Seria
Utilidad del programa.	Ocasional	Catastrófica
Acceso a la información.	Frecuente	Seria



## ESTRATEGIAS

Riesgo	Estrategia
Falta de trabajo de algún miembro del equipo.	Reorganización del trabajo asignado a cada miembro del equipo para que se redistribuyan las tareas convenientemente.
Incompatibilidad de hardware.	Reducción de los requisitos para ejecutar el programa y maximizar la compatibilidad con él.
Cambio en los requisitos.	Intentar aplicar el principio de la abstracción, en la medida de lo posible, de forma que cada una de las partes sea independiente a las otras y no altere en gran medida un cambio en alguna de las partes.
Retraso en el cumplimiento de los plazos.	Reorganizar el trabajo y el tiempo a dedicar por parte de cada miembro del grupo.
Compatibilidad con el sistema operativo	Alertar al cliente de los requisitos que el programa necesita.
Competencia.	Realizar una búsqueda sobre otros programas disponibles que tengan relación con la aplicación en desarrollo para comparar funciones.
Coordinación.	Elaborar un documento especificando la tarea de cada miembro del equipo, ciñéndose estrictamente a dicha labor.
Falta de conocimiento.	Dedicar un tiempo a informarse y aprender a utilizar los recursos con los que vamos a contar a la hora de desarrollar el proyecto
Librerías.	Cambiar la especificación que habíamos hecho en un principio para adaptarla a la tecnología disponible
Almacenamiento de datos.	Pensar varias formas para almacenar los datos, de manera que tengamos otra solución en caso de fallo.
Introducción de datos.	Pensar varias formas de realizar esta tarea de forma que tengamos estrategias alternativas en caso de problemas.
Utilidad del programa.	Hacer un estudio de mercado, para ver si hay demanda de un programa con las características del que estamos desarrollando, así como, si esa demanda está satisfecha o va a ser satisfecha en un corto plazo de tiempo, de manera que observemos si resulta rentable o no, invertir tiempo y dinero en este proyecto.
Acceso a la información.	El uso de la información del programa se restringirá al usuario que lo haya adquirido, evitando así que terceros puedan manipular la información de la base de datos.

---

## 5. PLANIFICACIÓN

---

---

### INICIO

---

**CAPTURA DE REQUISITOS:** Obtener una idea general del proyecto y analizar los requisitos necesarios para realizar el proyecto.

**Tiempo:** 1 mes.

**Fecha prevista:** 20 de Diciembre de 2011.

---

### ELABORACIÓN

---

**ANÁLISIS Y DISEÑO:** Planificar la estructura de datos, la interfaz y el funcionamiento del programa.

**Tiempo:** 1-2 meses.

**Fecha prevista:** 20 de Febrero de 2012.

---

### CONSTRUCCIÓN

---

**IMPLEMENTACIÓN:** Elaboración del programa en el lenguaje de programación requerido.

**Tiempo:** 1-2 meses.

**Fecha prevista:** 15 de Abril de 2012.

---

### TRANSICIÓN

---

**TESTEO Y DEPURACIÓN:** Realizar pruebas para comprobar el correcto funcionamiento del programa y corregir los errores que aparezcan, así como optimizar el programa.

**Tiempo:** 1 mes.

**Fecha prevista:** 20 de Mayo de 2012.