




INFORME FINAL TFC

DESARROLLO DE UNA BASE DE DATOS Y UNA APLICACIÓN WEB DE ACCESO A LA MISMA PARA UN GRUPO DE INVESTIGACIÓN MARINA DE ORGANISMOS DE PROFUNDIDAD (DEEP-SEA).

Autor: José Antonio García del Arco

Tutor UOC: Ferran Prados Carrasco 
www.uoc.edu

Tutor CSIC: Joan Baptista Company Claret   **CSIC**
CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

AGRADECIMIENTOS

Primero de todo agradecer al Instituto de Ciencias del Mar del Consejo Superior de Investigaciones Científicas, institución para la cual trabajo, la posibilidad de desarrollar este trabajo dentro de la misma.

Este trabajo tampoco hubiese sido posible sin la financiación ofrecida por el estado español y la unión europea a través de los proyectos PROMETEO (CTM2007-66316-C02-01/MAR) y HERMINONE (FP7-ENV-2008-1-226354), respectivamente.

Ahora podría mencionar a todos los que en algo han ayudado a terminar esta tarea, pero seguro me dejaría a alguno. A todos ellos mi más sincero agradecimiento.

Sin embargo, no puedo dejar de destacar a Ferran Prados tutor de la Universitat Oberta de Catalunya, por su apoyo y por aceptar este trabajo dentro de su área de formación.

Ni tampoco al Dr. Joan Baptista Company, por creer que los datos se pueden gestionar de otra manera y por buscar la financiación para que este trabajo haya ido adelante. ¡Ya casi estamos!

Así mismo me gustaría mencionar al desarrollador de *Drupal* Wim Leers, por hacer comprensibles con sus tutoriales el *JavaScript* y la tecnología *AHAH*, sin la cual una gran parte de este trabajo no se hubiera podido realizar.

No quiero olvidar al Dr. Jacopo Aguzzi, que me ha ocasionado y me ocasiona, no pocos quebraderos de cabeza, a él decirle que seguiremos en la brecha.

También a mi “despacho”, que en parte han sido acompañantes en el transcurso del conjunto del trabajo aquí presentado.

Por último dejo a mi familia, los grandes perjudicados de esta aventura en que me metí ya hace cinco años, sufridores de mis nervios y malhumor cuando las cosas no salen bien.

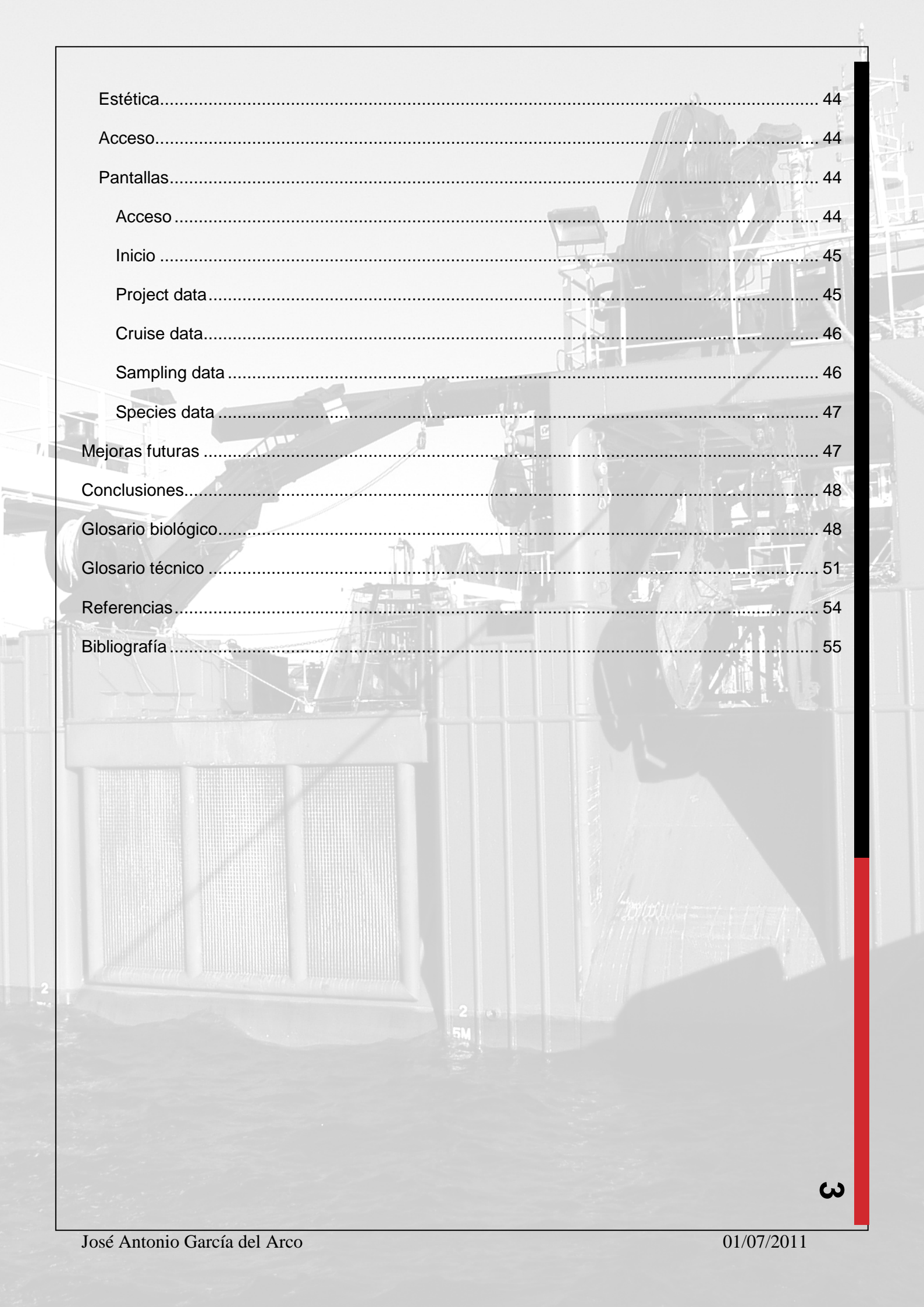
Mis agradecimientos sinceros a todos ellos y a todos los que en un momento u otro me facilitaron su apoyo.

¹ Fotografía de portada realizada durante la expedición oceanográfica del proyecto [BIOFUN](#) en junio del 2009. Autor: Dr. Matthias Schneidern de Senckenberg world of biodiversity: Research Institute and Natural History Museum Frankfurt ⁽¹⁾. Especie: *Chauliodus sloani*.



ÍNDICE

Agradecimientos	1
Índice	2
Introducción	4
Objetivos del trabajo.....	4
Prioritarios	4
Secundarios.....	5
Fases del trabajo.....	5
Plan de trabajo	5
Relación de tareas a realizar.....	5
Diagrama de Gantt.....	7
Análisis y especificaciones	8
Límites y marco del problema.....	8
Análisis de requerimientos.....	12
Actores implicados y tareas que desarrollan.....	14
Descripción de subsistemas.....	18
Diseño	18
Diagramas de actividad.....	18
Modelo de la Web.....	21
Modelo de datos implementado.....	30
Implementación	33
Tecnologías utilizadas.....	33
Entorno de programación.....	33
Gestor de contenido.....	33
Módulos de programación desarrollados.....	39
Descripción del web	44



Estética.....	44
Acceso.....	44
Pantallas.....	44
Acceso	44
Inicio	45
Project data.....	45
Cruise data.....	46
Sampling data	46
Species data	47
Mejoras futuras	47
Conclusiones.....	48
Glosario biológico.....	48
Glosario técnico	51
Referencias.....	54
Bibliografía	55

INTRODUCCIÓN

Este proyecto se fraguó previamente a la elección de un trabajo final de carrera. Para poder entenderlo debo mencionar que trabajo como titulado superior de investigación y laboratorio en el Instituto de Ciencias del Mar (ICM)⁽²⁾ del Consejo Superior de Investigaciones Científicas (CSIC)⁽³⁾, dentro de un grupo de investigación en oceanografía biológica, concretamente en recursos marinos renovables. En base a mi experiencia con este tipo de entorno de investigación, observe que existían una serie de mejoras de carácter técnico que se podrían introducir, y que a la larga iban a facilitar mucho más el trabajo científico del grupo.

Este grupo durante muchos años se ha dedicado a la obtención de datos de dos especies marinas de interés comercial del mar Mediterráneo que tienen su hábitat en aguas profundas: la gamba rosada (*Aristeus antennatus*) y la cigala (*Nephrops norvegicus*). Por ende, de manera colateral, datos de las especies que interaccionan con ellas, y que por este hecho se ven influenciadas al ser pescadas las anteriores. En estos años ha ido en aumento la evidencia de que ecosistemas más someros de nuestros mares tienen una relación mucho mayor de lo que se suponía con los ecosistemas profundos de los mismos. Además estos ecosistemas profundos influyen en los someros, también más de lo que cabía esperar, actuando de refugio de larvas y especies que tienen capacidad de sobrevivir en rangos batimétricos amplios. Si desean tener una visión más profunda al respecto pueden ver algunas de las últimas referencias bibliográficas a las que hago referencia en este párrafo acerca de este hecho, así como del incremento de la importancia de los grupos de investigación en el mundo dedicados a este tipo de investigación. En algunas de estas publicaciones han participado miembros del grupo al cual va dirigido el trabajo que aquí expongo⁽⁴⁾.
(5), (6), (7), (8), (9), (10), (11).

A medida que crecía el número de miembros del grupo, la importancia del mismo, la mejora tecnológica empleada en los muestreos, las colaboraciones internacionales con otras instituciones y la cantidad de proyectos en el grupo de investigación, crecía a su vez proporcionalmente, la cantidad de datos y la disparidad en formatos y sistemas de almacenaje (Hojas MS Excel⁽¹²⁾ o bases de datos MS Access⁽¹³⁾, archivos de texto, etc.). Se ha hecho necesaria entonces la creación de una herramienta que los gestione de una forma común, y una base de datos para el almacenaje de los mismos de una forma coherente y robusta. Así mismo el hecho de tener los datos en una fuente común, posibilitará su distribución a otras bases de datos mundiales sobre la materia⁽¹⁴⁾,⁽¹⁵⁾ con las cuales se colabora, dependientes de organismos tan en la cresta a de la ola, como el *Census of marine life (COML)*, el *Alfred Wegener Institute (AWI)* y el *Center for Marine Environmental Sciences (MARUM)*⁽¹⁶⁾,⁽¹⁷⁾,⁽¹⁸⁾ de Alemania. Estos a su vez carecen de datos de las zonas geográficas pertenecientes al mar Mediterráneo foco de la investigación del grupo.

Finalmente me gustaría reseñar que este documento está plagado de términos técnicos provenientes de dos campos tan diferentes como son la biología pesquera y la informática. A fin de hacer mucho más inteligible el texto al lector he creado un extenso glosario (al final del documento), en el que espero halle resueltas todas sus dudas en lo que a terminología se refiere.

OBJETIVOS DEL TRABAJO

Prioritarios

- Estandarizar los datos en unidades, orden de los campos de almacenaje y tipología. Existen datos numéricos que varían en unidades, precisión y disposición de las columnas diversas. Ocurre lo mismo con los datos de tipo carácter hay una gran variedad y nomenclaturas diferentes (en parte realizado ya).
- Crear una base de datos de acceso común que los aglutine los mismos. Para ello se ha propuesto y de hecho ya se está utilizando el sistema de gestión de bases de datos relacionales de software libre MySQL⁽¹⁹⁾.

- Crear una aplicación Web amigable, consensuada con los investigadores, de acceso único a los datos.
- Crear formularios de introducción de datos únicos y estándar que puedan ser usados en el transcurso de las expediciones oceanográficas, evitando la disgregación actual de los mismos y facilitando un mayor control de calidad de los mismos. Su diseño al igual que el de la aplicación Web también deberá ser consensuado.

Secundarios

- Implementar utilidades de exportación de los mismos a formato de MS Excel ⁽¹²⁾ o texto plano.
- Posibilitar el filtrado, la transformación y realización de algunos cálculos estadísticos sobre los datos almacenados según necesidades del investigador.
- Posibilitar el acceso a los datos desde otras fuentes de datos relacionadas con el ámbito de investigación de otros organismos de investigación internacionales.
- Poder realizar el volcado de los datos anteriores en el transcurso de las expediciones oceanográficas, utilizando para ello el acceso a Internet mundial del que disponen actualmente los buques oceanográficos pertenecientes al CSIC ⁽³⁾.

FASES DEL TRABAJO

Plan de trabajo

Habitualmente en este tipo de proyectos es común presentar retrasos. En el caso de este trabajo los retrasos más destacables se han dado en las tareas de implementación en conjunción con la de documentación. A medida que se avanzaba en la implementación aparecían nuevos retos que debían de solucionarse. O bien buscando una solución previa realizada por algún desarrollador, o bien buscando una tecnología que nos permitiese implementar la funcionalidad que nosotros deseábamos. Los documentos y las Web para documentarse ya eran conocidos de las fases previas del proyecto, el problema viene cuando se parte de cero, y se hace necesario el entendimiento de nuevos lenguajes de programación y tecnologías, como ha ocurrido en las tareas de implementación que menciono.

Así las tareas que han presentado un mayor retraso y que todavía se hallan en desarrollo son las de creación de los formularios de entrada de datos y de salida de los mismos. Hallándose en estos momentos en un 50% de cumplimiento.

RELACIÓN DE TAREAS A REALIZAR.

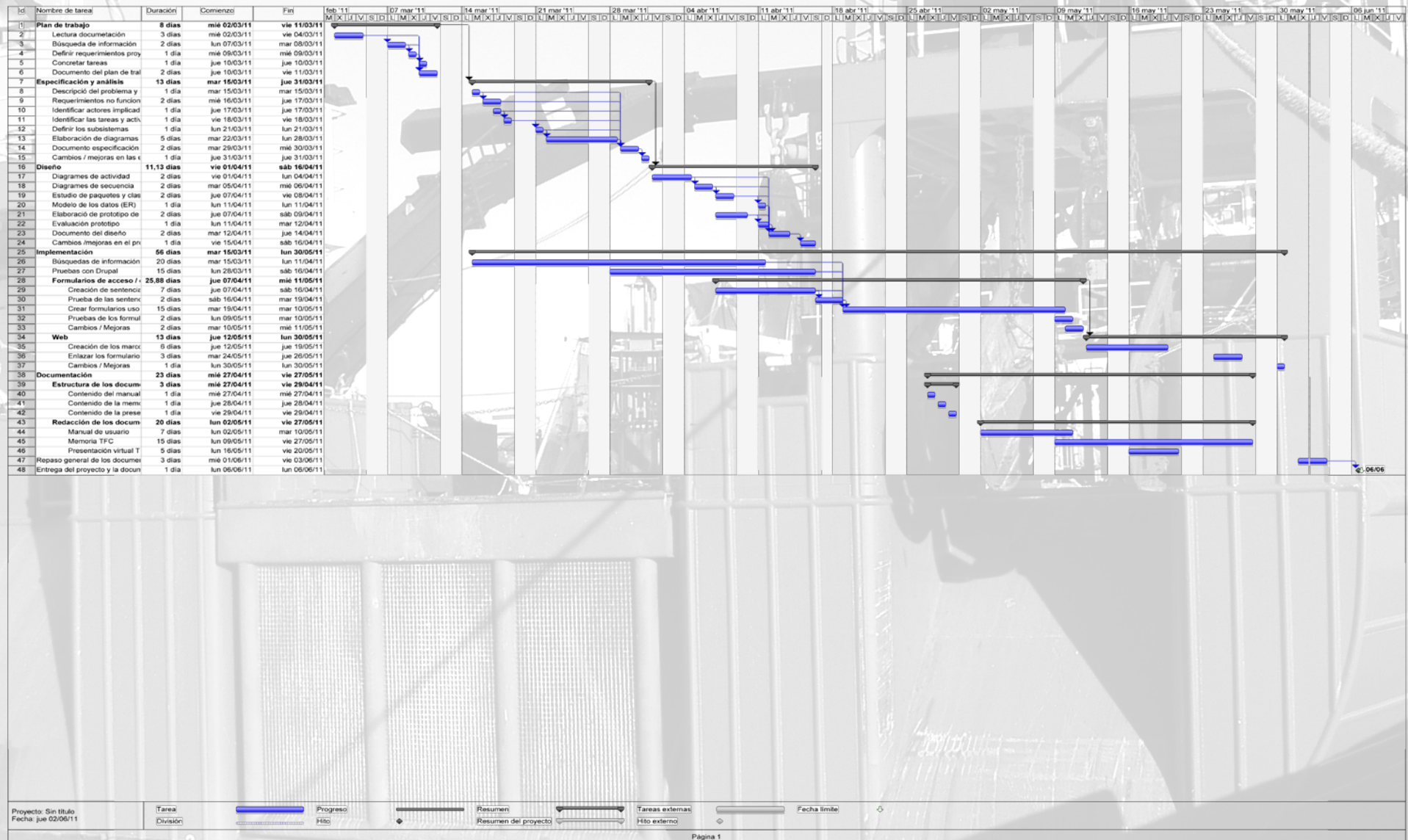
1.	Plan de trabajo	8d	mié 02/03/11	vie 11/03/11
2.	Lectura documentación	3d	mié 02/03/11	vie 04/03/11
3.	Búsqueda de información	2d	lun 07/03/11	mar 08/03/11
4.	Definir requerimientos proyecto	1d	mié 09/03/11	mié 09/03/11
5.	Concretar tareas	1d	jue 10/03/11	jue 10/03/11
6.	Documento del plan de trabajo	2d	jue 10/03/11	vie 11/03/11
7.	Especificación y análisis	13d	mar 15/03/11	jue 31/03/11
8.	Descripción del problema y su marco	1d	mar 15/03/11	mar 15/03/11
9.	Requerimientos no funcionales	2d	mié 16/03/11	jue 17/03/11
10.	Identificar actores implicados	1d	jue 17/03/11	jue 17/03/11
11.	Identificar las tareas y actividades de cada actor	1d	vie 18/03/11	vie 18/03/11
12.	Definir los subsistemas	1d	lun 21/03/11	lun 21/03/11
13.	Elaboración de diagramas y casos de uso	5d	mar 22/03/11	lun 28/03/11
14.	Documento especificación y análisis	2d	mar 29/03/11	mié 30/03/11
15.	Cambios / mejoras en las especificaciones	1d	jue 31/03/11	jue 31/03/11
16.	Diseño	11,13d	vie 01/04/11	sáb 16/04/11
17.	Diagramas de actividad	2d	vie 01/04/11	lun 04/04/11

- | | | | | | |
|-----|--|--------|--------------|--------------|--|
| 18. | Diagramas de secuencia | 2d | mar 05/04/11 | mié 06/04/11 | |
| 19. | Estudio de paquetes y clases (Diagramas) | 2d | jue 07/04/11 | vie 08/04/11 | |
| 20. | Modelo de los datos (ER) | 1d | lun 11/04/11 | lun 11/04/11 | |
| 21. | Elaboración de prototipo de la WEB | 2d | jue 07/04/11 | sáb 09/04/11 | |
| 22. | Evaluación prototipo | 1d | lun 11/04/11 | mar 12/04/11 | |
| 23. | Documento del diseño | 2d | mar 12/04/11 | jue 14/04/11 | |
| 24. | Cambios /mejoras en el prototipo y / o diseño | 1d | vie 15/04/11 | sáb 16/04/11 | |
| 25. | Implementación | 56d | mar 15/03/11 | lun 30/05/11 | |
| 26. | Búsquedas de información y estudio de manuales | 20d | mar 15/03/11 | lun 11/04/11 | |
| 27. | Pruebas con Drupa | 15d | lun 28/03/11 | sáb 16/04/11 | |
| 28. | Formularios de acceso / escritura de datos | 25,88d | jue 07/04/11 | mié 11/05/11 | |
| 29. | Creación de sentencias y consultas SQL | 7d | jue 07/04/11 | sáb 16/04/11 | |
| 30. | Prueba de las sentencias y consultas | 2d | sáb 16/04/11 | mar 19/04/11 | |
| 31. | Crear formularios uso del API de Drupal | 15d | mar 19/04/11 | mar 10/05/11 | |
| 32. | Pruebas de los formularios | 2d | lun 09/05/11 | mar 10/05/11 | |
| 33. | Cambios / Mejoras | 2d | mar 10/05/11 | mié 11/05/11 | |
| 34. | Web | 13d | jue 12/05/11 | lun 30/05/11 | |
| 35. | Creación de los marcos | 6d | jue 12/05/11 | jue 19/05/11 | |
| 36. | Enlazar los formularios | 3d | mar 24/05/11 | jue 26/05/11 | |
| 37. | Cambios / Mejoras | 1d | lun 30/05/11 | lun 30/05/11 | |
| 38. | Documentación | 23d | mié 27/04/11 | vie 27/05/11 | |
| 39. | Estructura de los documentos | 3d | mié 27/04/11 | vie 29/04/11 | |
| 40. | Contenido del manual de usuario | 1d | mié 27/04/11 | mié 27/04/11 | |
| 41. | Contenido de la memoria TFC | 1d | jue 28/04/11 | jue 28/04/11 | |
| 42. | Contenido de la presentación virtual | 1d | vie 29/04/11 | vie 29/04/11 | |
| 43. | Redacción de los documentos | 20d | lun 02/05/11 | vie 27/05/11 | |
| 44. | Manual de usuario | 7d | lun 02/05/11 | mar 10/05/11 | |
| 45. | Memoria TFC | 15d | lun 09/05/11 | vie 27/05/11 | |
| 46. | Presentación virtual TFC | 5d | lun 16/05/11 | vie 20/05/11 | |
| 47. | Repaso general de los documentos | 3d | mié 01/06/11 | vie 03/06/11 | |
| 48. | Entrega del proyecto y la documentación | 1d | lun 06/06/11 | lun 06/06/11 | |

Todas aquellas tareas relacionadas con los objetivos secundarios no se han tenido en cuenta en esta programación, ya que se ha considerado que la carga de trabajo era superior a la que se podría desarrollar en este periodo de tiempo.

DIAGRAMA DE GANTT

Figura 1.



Análisis y especificaciones

LÍMITES Y MARCO DEL PROBLEMA.

Primeramente tengo que destacar que todo tipo de acción (campañas oceanográficas, experimentos o este mismo proyecto final de carrera) que se llevan a cabo, van asociados a un determinado proyecto de investigación. Este tiene un investigador /es principal /es y su financiación proviene de diferentes ámbitos (local, regional, nacional o europeo).

Un muestreo o campaña oceanográficos suponen un gran despliegue de medios humanos, técnicos y financieros. Durante el transcurso de la misma se adquieren un sinfín de datos de diferente cariz: Físico, biológico o geológico. Nuestra solución sólo se centrará en su aspecto biológico y todo lo relacionado con el mismo, pero que es aplicable en estructura al resto de tipología de datos. Siendo factible la posibilidad de ampliar esta solución a todo tipo de datos que se producen en campaña, mediante la característica de importación masiva de datos, contemplada como un objetivo secundario en el apartado de objetivos del trabajo. El hecho de que los datos de carácter físico o geológico sean de gran volumen y en su gran mayoría proporcionados por sistemas automáticos de adquisición de datos (que no describiremos en este documento, ya que se alejan de los objetivos primarios) hace que la importación masiva de los mismos sea la forma más adecuada de introducirlos en el sistema.

Para una campaña oceanográfica será necesario almacenar los datos relativos al proyecto que la financia, el investigador principal que la dirigirá, el buque oceanográfico que la llevará a cabo, y todos los muestreos que se han realizado durante la misma.

En el transcurso de una campaña oceanográfica (centrados en el problema biológico) los datos comunes que se deben de conocer para cada maniobra de muestreo o recolección de especímenes son:

- ✓ Código distintivo único.
- ✓ Área geográfica del muestreo.
- ✓ Área geomorfológica del muestreo.
- ✓ Tipo de instrumento oceanográfico.
- ✓ Fecha de inicio del muestreo.
- ✓ Fecha de fin del muestreo.
- ✓ Profundidad en el punto de muestreo
- ✓ Hora de inicio del muestreo.
- ✓ Hora final del muestreo.

Posteriormente dependiendo de la tipología del aparato de muestreo tendremos datos diferentes a almacenar:

A. Sistemas en que se muestrea de forma lineal:

a. OTMS (Otter Trawl Maireta System ©):

Por una parte están los datos que se relacionan con la maquinilla del barco:

- ✓ Hora de parada de la maquinilla del barco.
- ✓ Latitud en el momento de la parada de la maquinilla.
- ✓ Longitud en el momento de la parada de la maquinilla.
- ✓ Hora del inicio del virado de la maquinilla.
- ✓ Latitud en el momento de virar.
- ✓ Longitud en el momento de virar.
- ✓ Velocidad media del barco durante la maniobra de muestreo.
- ✓ Longitud del cable de maquinilla utilizado.
- ✓ Abertura horizontal de la boca del arte de pesca (dato teórico).
- ✓ Tiempo de pesca (calculado).
- ✓ Área barrida durante la maniobra por el arte de pesca (calculado).



- ✓ Peso total de lo capturado por la red.
- ✓ Peso total de los animales capturados por la red.
- ✓ Peso total de la basura en la captura.
- ✓ Observaciones varias.

Después están los datos que facilitan los sensores del *Scanmar*, que no siempre se tienen o son fiables por diferentes motivos y problemas técnicos:

- ✓ Hora de contacto de la red con el fondo.
- ✓ Latitud en el momento del contacto.
- ✓ Longitud en el momento del contacto.
- ✓ Hora del inicio del despegue de la red del fondo.
- ✓ Latitud en el momento del despegue.
- ✓ Longitud en el momento del despegue.

Otros datos a tener en cuenta que nos facilitan los sensores *Scanmar* a lo largo del tiempo de la pesca:

- ✓ Hora de la toma del dato.
- ✓ Altura de red en ese momento.
- ✓ Anchura de red en ese momento.

b. Patín Agassiz:

- ✓ Hora de parada de la maquinilla del barco.
- ✓ Latitud en el momento de la parada de la maquinilla.
- ✓ Longitud en el momento de la parada de la maquinilla.
- ✓ Hora del inicio del virado de la maquinilla.
- ✓ Latitud en el momento de virar.
- ✓ Longitud en el momento de virar.
- ✓ Velocidad media del barco durante la maniobra de muestreo.
- ✓ Longitud del cable de maquinilla utilizado.
- ✓ Abertura horizontal de la boca del arte de pesca (dato teórico).
- ✓ Tiempo de pesca (calculado).
- ✓ Área barrida durante la maniobra por el arte de pesca (calculado).
- ✓ Peso total de lo capturado por la red.
- ✓ Peso total de los ejemplares.
- ✓ Peso total de la basura en la captura.
- ✓ Observaciones varias.

c. Epibenthic Sled:

- ✓ Hora de parada de la maquinilla del barco.
- ✓ Latitud en el momento de la parada de la maquinilla.
- ✓ Longitud en el momento de la parada de la maquinilla.
- ✓ Hora del inicio del virado de la maquinilla.
- ✓ Latitud en el momento de virar.
- ✓ Longitud en el momento de virar.
- ✓ Velocidad media del barco durante la maniobra de muestreo.
- ✓ Longitud del cable de maquinilla utilizado.
- ✓ Tiempo de pesca (calculado).
- ✓ Lectura inicial *Flowmeter* compartimento superior.
- ✓ Lectura inicial *Flowmeter* compartimento medio.
- ✓ Lectura inicial *Flowmeter* compartimento inferior.
- ✓ Lectura final *Flowmeter* compartimento superior.
- ✓ Lectura final *Flowmeter* compartimento medio.
- ✓ Lectura final *Flowmeter* compartimento inferior.
- ✓ Volumen filtrado en el compartimento superior (calculado).
- ✓ Volumen filtrado en el compartimento medio (calculado).

- ✓ Volumen filtrado en el compartimiento inferior (calculado).
- ✓ Peso total de los individuos capturados en el compartimiento superior.
- ✓ Peso total de los individuos capturados en el compartimiento medio
- ✓ Peso total de los individuos capturados en el compartimiento inferior
- ✓ Observaciones varias

d. Rectangular Midwater Trawl, RMT:

- ✓ Hora de inicio maniobra.
- ✓ Hora de fin de maniobra.
- ✓ Latitud en el momento de echar la red al agua.
- ✓ Longitud en el momento de echar la red al agua.
- ✓ Latitud en el momento de sacar la red del agua.
- ✓ Longitud en el momento de sacar la red del agua.
- ✓ Observaciones varias.

Esta red está dividida en dos de las que a su vez se requieren los datos que siguen:

- ✓ Profundidad inicial.
- ✓ Profundidad final.
- ✓ Metros de cable de la maquinilla.
- ✓ Duración del muestreo.

e. Isaacs-Kidd Mid-water Trawl, IKMT:

- ✓ Hora de inicio maniobra.
- ✓ Hora de fin de maniobra.
- ✓ Latitud en el momento de echar la red al agua.
- ✓ Longitud en el momento de echar la red al agua.
- ✓ Latitud en el momento de sacar la red del agua.
- ✓ Longitud en el momento de sacar la red del agua.
- ✓ Profundidad del inicio de despliegue de la red.
- ✓ Profundidad del final de despliegue de la red.
- ✓ Longitud del cable usado en el despliegue de la red.
- ✓ Duración de la maniobra de despliegue.
- ✓ Profundidad del inicio de pesca.
- ✓ Profundidad del final de pesca.
- ✓ Longitud del cable usado en la pesca.
- ✓ Duración de la pesca.
- ✓ Observaciones varias.

B. Sistemas de muestreo en un punto:

a. Multicorer:

- ✓ Latitud del punto de muestreo.
- ✓ Longitud del punto de muestreo.
- ✓ Observaciones varias.

Del *Multicorer* existen datos específicos de carácter biológico y geológico diferentes. Para los geológicos se toman los datos que siguen, aparte de todos los demás comunes, pero que no tendremos en cuenta en este proyecto por pertenecer a un objetivo secundario. Sólo los detallo a modo de ejemplo de lo que comento al inicio de este punto:

- ✓ Código distintivo del núcleo.
- ✓ Longitud del núcleo.
- ✓ Identificador de la persona que muestrea.
- ✓ Centímetro muestreado.
- ✓ Volumen muestreado.
- ✓ WDB.

- ✓ DBD.
- ✓ Contenido de agua.
- ✓ Porosidad.
- ✓ Porcentaje total de carbono contenido en la muestra.
- ✓ Porcentaje de carbono orgánico en la muestra.
- ✓ Porcentaje de carbono inorgánico en la muestra.
- ✓ Porcentaje de carbonato de calcio en la muestra.
- ✓ Media.
- ✓ Mediana.
- ✓ Moda.
- ✓ Porcentaje de arcilla.
- ✓ Porcentaje de sílice.
- ✓ Porcentaje de arena.

Para los biológicos se deben almacenar:

- ✓ Código distintivo del núcleo.
- ✓ Peso total de los individuos capturados.

Las capturas obtenidas mediante estos muestreos se deben dividir en lotes de los que necesitamos conocer la su código, categoría biológica, género y especie. De cada uno de los lotes debemos conocer, a su vez, el número de individuos que compone el lote, su peso y observaciones varias acerca del lote.

Finalmente de los lotes de las capturas obtenidas es necesario conocer para cada individuo, aparte de los datos de su lote (código del lote, categoría, género y especie):

- ✓ Código identificador único (facilitar el acceso y modificación de los datos).
- ✓ Sexo.
- ✓ Talla (en milímetros o centímetros).
- ✓ Peso (gramos o miligramos).
- ✓ Estado de reproducción.
- ✓ Observaciones acerca del individuo.

Reseñar a su vez que un individuo sólo podrá tener una categoría, género y especie. Pero que una categoría estará compuesta de uno o varios géneros, y a su vez estos estarán compuestos de una o varias especies.

Finalmente los individuos dependiendo de su categoría será necesario introducir unos datos concretos. Por ahora sólo usaremos tres tipologías para estos: Peces, crustáceos y otros.

Para peces se necesitará conocer:

- ✓ Longitud total del individuo (milímetros o centímetros depende de la especie).
- ✓ Estado de reproducción modificado.
- ✓ Peso gonadal.
- ✓ Índice GSI.
- ✓ Histología realizada.
- ✓ Numero de porta.
- ✓ Extracción de otolitos.

Para crustáceos deberemos almacenar:

- ✓ Presencia de espermátforo.
- ✓ Estado de los huevos.
- ✓ Parásitos (Sacculinas y bopíridos).
- ✓ Estado de muda.

Respecto a la categoría "otros" queda contemplada en los datos recogidos al nivel de individuo.

Con todo esta relación de aparatos de muestreo y datos queda claramente reflejado que la complejidad de este proyecto radica en el volumen de la tipología de los datos y en la correcta comprensión de la intrincada relación existente en los mismos. Ya que en resumen será un portal de entrada de datos común a los miembros del grupo de investigación para los cuales va dirigido este proyecto.

Por último quedaría explicar las posibles salidas de datos que los usuarios demandan, y que he reducido inicialmente a tres tipos de listados de datos diferentes.

- ✓ El primero de estos listados en su versión más simple es una lista donde aparezca: proyecto, campaña, fecha, código de muestreo, categoría, género, especie, observaciones, número de individuos y peso (normalizados por área de barrido de la red, estos dos últimos). Estos se deben poder filtrar y agrupar por todos sus campos menos los tres últimos.
- ✓ Por otro lado también se podrán generar listas con los datos de proyecto, código de muestreo, coordenadas, (iniciales y finales si es un muestreo de tipo lineal), fecha, hora, área barrida (si es un muestreo de tipo lineal), observaciones o el resto de datos de los otros tipos de muestreo según su tipología, de estos los principales a mostrar (por separado) son: Pescas de fondo, patín suprabentónico, patín Agassiz y el multicorer.
- ✓ El último tipo de listado a generar será un listado de los individuos capturados con los datos del código de individuo, proyecto, campaña, código de muestreo, fecha, categoría, género, especie, talla, peso, estado de reproducción y observaciones. Deben tener la posibilidad de filtrarlos por fecha, categoría género, especie, código de muestreo, campaña. Además también se deberán poder agrupar por rangos de talla, de profundidad, fecha, categoría, género y especie.

Para concluir este apartado sólo recordar que para estos datos, sobre todo en los del último tipo de listado en futuras ampliaciones de esta aplicación se deberá poder realizar determinadas estadísticas en los mismos, contempladas en el apartado de objetivos secundarios de este documento.

ANÁLISIS DE REQUERIMIENTOS.

Rendimiento del software.

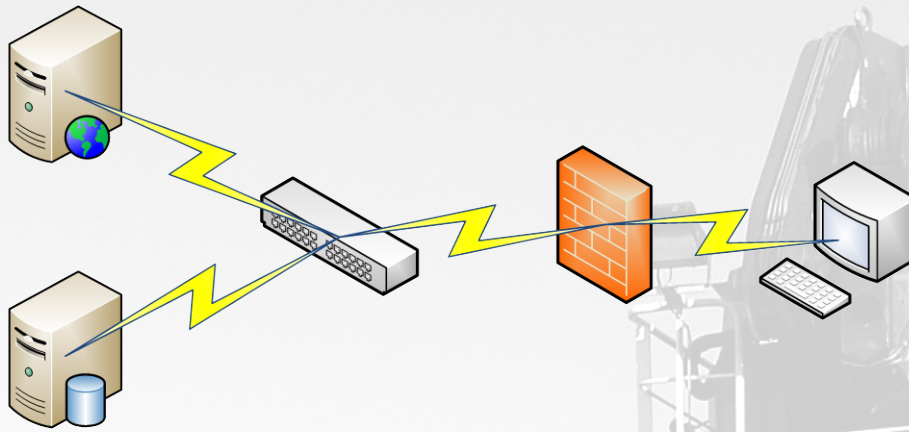
Los requerimientos de rendimiento para el software a desarrollar no son muy exigentes en cuanto a velocidad de ejecución, debido principalmente al bajo número de usuarios concurrentes que tendrá la aplicación. Si en cambio deben ser altos en robustez y facilidad de uso como consecuencia del tipo de usuarios que lo manejarán, son exigentes en cuanto a los fallos y buscan una gran facilidad de manejo en el mismo (ver apartado de usabilidad).

Las herramientas elegidas para el desarrollo de la aplicación y la base de datos (*Drupal*⁽²⁰⁾ y *MySQL*⁽¹⁹⁾) llevan intrínsecas estos requerimientos de robustez y rendimiento, por lo que estas características dependerán en gran parte de un diseño apropiado de la Web y la base de datos, o sea del diseñador y el programador.

Distribución del software.

El modelo de distribución del software será el clásico para las aplicaciones Web: Modelo Cliente-Servidor. Véase en la figura que sigue.

Figura 2.



En nuestro caso los servidores Web y de Base de Datos están ubicados en el mismo nodo físico, ya que se corresponde a dos máquinas virtuales con sistema operativo *Linux Ubuntu 10.04*⁽²¹⁾ ejecutadas en *VMware Player*⁽²²⁾ versión 3.0.1, que hacen de servidores. Están ubicadas en un Host provisional *Windows XP*⁽²³⁾.

Seguridad del software.

La seguridad de la aplicación viene determinada por la gran capacidad que tiene *Drupal*⁽²⁰⁾ para gestionar estos aspectos. Para lograrlo posee diferentes características:

- ✓ Control sobre la propiedad y permisos de los archivos.
- ✓ Posibilidad de activación del protocolo *HTTP Secure (HTTPS)*.
- ✓ Posibilidad de mejorar la seguridad añadiendo módulos contribuidos del mismo *Drupal*⁽²⁰⁾.
- ✓ Ocultación de la característica de que un Web se ejecuta en *Drupal*⁽²⁰⁾.
- ✓ Prevenir la ejecución de código *PHP*⁽²⁷⁾ sin verificar.
- ✓ Ocultar la información sensible a simples visitantes.
- ✓ Control de usuarios.

Por otro lado la seguridad en la transmisión de los datos se implementara mediante el uso del protocolo *SSH*, propiedad que viene implementada de origen en *Drupal*⁽²⁰⁾ y *MySQL*⁽¹⁹⁾.

Finalmente, la seguridad de la base de datos está garantizada por los estándares de las bases de datos relacionales que posee *MySQL*⁽¹⁹⁾, como son control de usuarios, archivos log de acceso, archivos log de datos, posibilidad de replicación de los datos, etc.

Usabilidad del software.

Los requerimientos de usabilidad de la aplicación buscados por los usuarios son muy sencillos. Principalmente se basaran en una gran sencillez de diseño que no interfiera en la tarea principal de la aplicación, así como en una gran facilidad en su manejo. Estas dos características conllevaran el último y principal requerimiento de este apartado que será el uso intuitivo de la aplicación, sin necesidad de tener que hacer una formación previa al primer uso de la aplicación, o de manuales complejos de la misma.

Para poder implementar estos requerimientos de usabilidad he desarrollado un prototipo navegable de la Web, al cual se puede acceder mediante el enlace que sigue:

<http://norit.cmima.csic.es:5800/prometeodatabase/web BBDD.htm>

Dentro del modelo en el apartado *Data output* se simulan dos posibles salidas de datos. Una seleccionando del desplegable *Select Project* el proyecto PROMETO, y pulsando *ok* aparece un

posible listado de datos. La otra seleccionando en el desplegable *Select species*, la especie *Alepocephalus rostratus*, de nuevo pulsando *ok* aparece otro posible listado de datos.

Este modelo de interface está basado en los formularios (en papel) de recogida de datos en campaña, a los que ya están habituados los usuarios en lenguaje y estructura. La interface Web que se presenta al final de este documento ha variado en estilo simplificándolo, y adaptándolo a los que utiliza el sistema de gestor de contenidos elegido para realizar este trabajo.

ACTORES IMPLICADOS Y TAREAS QUE DESARROLLAN.

Relación de actores i tareas.

De las entrevistas realizadas al personal de investigación, he identificado tres tipos de actores que a continuación presento con la lista de tareas que cada uno de ellos puede realizar.

3.1.1. Administrador:

- ✓ Alta de nuevos usuarios incluidos dentro de uno de estos tipos de actor.
- ✓ Entrada de nuevos proyectos de investigación y modificación de los ya introducidos.
- ✓ Modificación y corrección de datos ya consolidados en la base de datos.
- ✓ Acceso a listados de datos genéricos y también los especiales.
- ✓ Importación masiva de datos.
- ✓ Entrada y corrección en tiempo de ejecución de datos biológicos.
- ✓ Navegación por todo el Web.

3.1.2. Usuario:

- ✓ Acceso a listados de datos genéricos y también los especiales.
- ✓ Entrada y corrección en tiempo de ejecución de datos biológicos.
- ✓ Navegación por los espacios Web permitidos.

3.1.3. Usuario Web:

- ✓ Acceso a listados de datos genéricos.
- ✓ Navegación por los espacios Web permitidos.

Diagramas i fichas de casos de uso.

Diagramas de los casos de uso.

Figura 3.

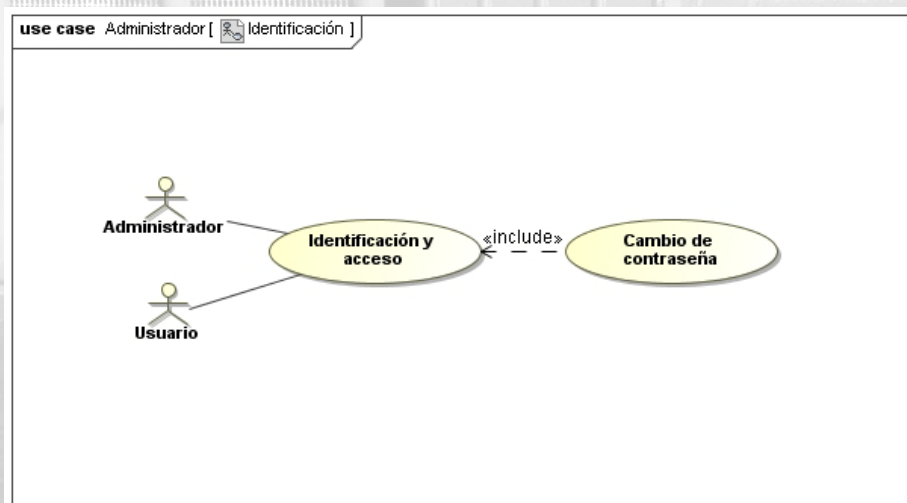


Figura 4.

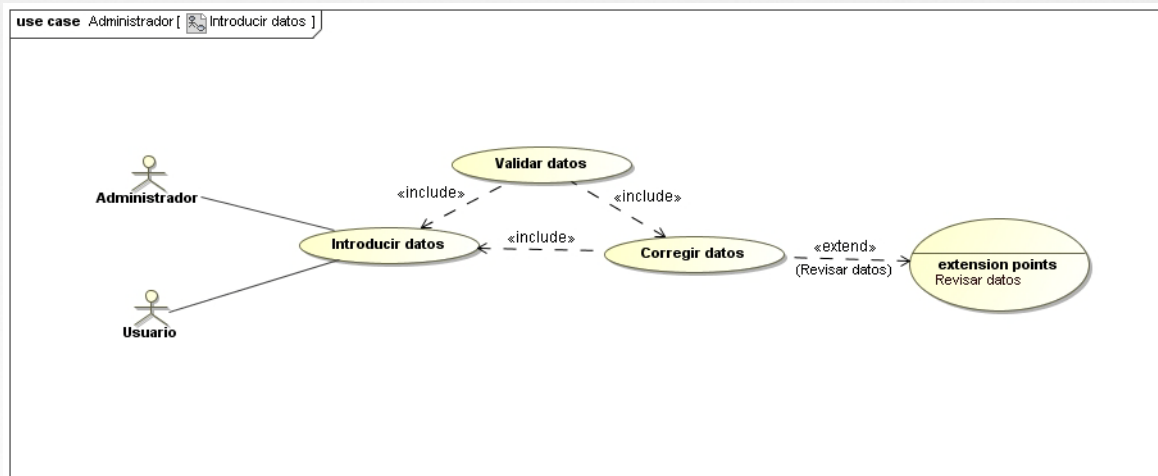
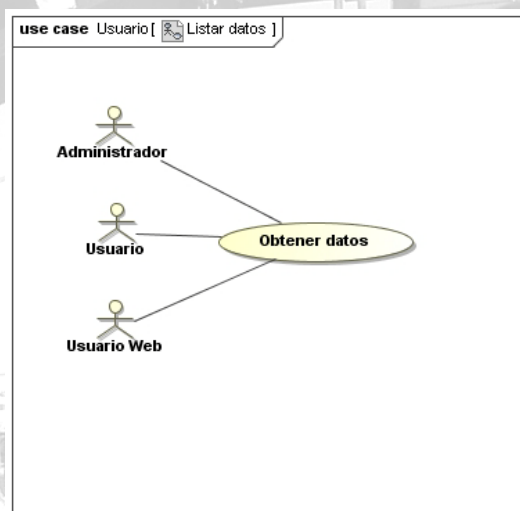


Figura 5.



Fichas de los casos de uso.

	IDENTIFICACIÓN
Resumen de funcionalidad	Identifica al usuario en el Web
Papel dentro del trabajo del usuario	Básico, se tiene que realizar cada vez que se utilice la aplicación para todas las tareas relacionadas con los datos.
Actores	Administrador y Usuario
Casos de uso relacionados	Cambiar la contraseña
Pre-condición	Usuario conoce y tiene los datos de conexión válidos
Post-condición	Usuario identificado y el usuario accede a las páginas a las que tiene permiso
Descripción	El sistema solicita el código de usuario y su contraseña. Comprueba la validez de los mismos, si es correcto le da acceso a las páginas web a las que tiene permisos
Alternativas de proceso y excepciones	Error de código de usuario Error de contraseña Error en ambos
Observaciones	La identificación se hará a petición del sistema cuando el usuario que navega quiera acceder a una página con privilegios. O también podrá ser a petición del usuario.

INTRODUCIR DATOS

Resumen de funcionalidad	Introducir datos en la base de datos
Papel dentro del trabajo del usuario	Prioritario para usuarios sin derecho de administración
Actores	Administrador y Usuario
Casos de uso relacionados	Validar datos, corregir datos y revisar datos
Pre-condición	Usuario está identificado y tiene los privilegios necesarios
Post-condición	Datos del formulario validados e introducidos en la base de datos
Descripción	El usuario con privilegios puede ir accediendo a las diferentes páginas e introducir los diferentes datos en la base de datos mediante el uso de los diferentes formularios de datos
Alternativas de proceso y excepciones	Error en los datos Correcciones en los datos Cancelación
Observaciones	Los datos son ignorados al hacer una cancelación. En este proceso no se habrá interactuado en ningún momento con la base de datos

OBTENER DATOS

Resumen de funcionalidad	Generar listados de datos
Papel dentro del trabajo del usuario	Opcional
Actores	Administrador y Usuario
Casos de uso relacionados	Exportar datos
Pre-condición	Usuario está identificado y tiene los privilegios necesarios
Post-condición	Se ha generado un listado de datos
Descripción	El usuario con privilegios puede ir accediendo a las diferentes páginas y realizar las peticiones para generar los listados de datos personalizados. Los podrá tener por pantalla o ser exportados a formato compatible con MS Excel ⁽¹²⁾
Alternativas de proceso y excepciones	Error de acceso a la base de datos
Observaciones	

EXPORTAR DATOS

Resumen de funcionalidad	Exportar los listados de datos
Papel dentro del trabajo del usuario	Opcional
Actores	Administrador y Usuario
Casos de uso relacionados	Obtener datos
Pre-condición	Se ha generado un listado de datos
Post-condición	Los datos de han exportado en un archivo compatible con MS Excel ⁽¹²⁾
Descripción	El usuario con privilegios ha generado los listados de datos personalizados y estos son exportados a formato compatible con MS Excel ⁽¹²⁾
Alternativas de proceso y excepciones	Error de escritura o acceso a disco
Observaciones	

CAMBIO DE CONTRASEÑA

Resumen de funcionalidad	Permite a los usuarios con contraseña su cambio
Papel dentro del trabajo del usuario	Opcional
Actores	Administrador y Usuario
Casos de uso relacionados	Identificación
Pre-condición	El usuario es válido y tiene contraseña
Post-condición	Contraseña cambiada
Descripción	Se accede a una página que permite a un usuario el cambio de contraseña. Este cambio puede ser a petición propia o hacerlo el administrador por algún requerimiento de seguridad. Para proceder al cambio el usuario que sea debe introducir la contraseña antigua y posteriormente la nueva contraseña dos veces. Finalmente si está de acuerdo proceder a aceptar el cambio, o sino cancelarlo
Alternativas de proceso y excepciones	Contraseña no válida
Observaciones	

CORREGIR DATOS

Resumen de funcionalidad	Realizar la corrección de líneas de datos sin enviar a la base de datos
Papel dentro del trabajo del usuario	Prioritario
Actores	Administrador y Usuario
Casos de uso relacionados	Introducir datos y revisar datos
Pre-condición	Usuario está identificado y tiene los privilegios necesarios
Post-condición	Datos del formulario corregidos y / o introducidos en la base de datos
Descripción	El usuario con privilegios puede ir accediendo a las diferentes páginas de introducción de datos. A medida que incluye datos aparecen las líneas introducidas debajo, seleccionando una de estas líneas se vuelven a editar en el formulario y permite su corrección
Alternativas de proceso y excepciones	
Observaciones	En este proceso no se habrá interactuado en ningún momento con la base de datos

REVISAR DATOS

Resumen de funcionalidad	Revisar y editar datos de la base de datos
Papel dentro del trabajo del usuario	Opcional
Actores	Administrador
Casos de uso relacionados	Introducir datos y validar datos
Pre-condición	Usuario está identificado y tiene los privilegios necesarios
Post-condición	Datos editados y / o corregidos y validados
Descripción	Esta tarea tiene lugar cuando se detecta un error en datos ya introducidos. Se debe facilitar la posibilidad de su edición por código de muestreo y facilitar su revisión y edición mediante el uso de los formularios de introducción de datos
Alternativas de proceso y excepciones	Error de acceso a la base de datos, puede ser de lectura

excepciones	o de escritura
Observaciones	En este proceso los datos son extraídos de la base de datos y se cargan en los mismos formularios de introducción de datos
VALIDAR DATOS	
Resumen de funcionalidad	Validan los datos y son introducidos en la base de datos de forma permanente
Papel dentro del trabajo del usuario	Prioritario
Actores	Administrador y Usuario
Casos de uso relacionados	Introducir datos, revisar datos y corregir datos
Pre-condición	Usuario está identificado y tiene los privilegios necesarios y hay un lote de datos para ser validados
Post-condición	Datos han sido validados e introducidos en la base de datos
Descripción	El usuario dentro de la página de introducción de datos ejecuta la validación que se hace de una manera automática y los datos se envían posteriormente a la base de datos, quedando el formulario de introducción de datos limpio
Alternativas de proceso y excepciones	Error de escritura en la base de datos
Observaciones	

DESCRIPCIÓN DE SUBSISTEMAS.

Para poder implementar este proyecto se ha tomado la decisión de dividirlo en varios subsistemas que gestionan tareas similares o iguales. Esta subdivisión es la que sigue: Conexión y acceso a las bases de datos, gestión de datos y estadísticas.

Conexión i acceso a las bases de datos.

Este subsistema será el encargado de la gestión de la conexión al servidor de la base de datos, del control de acceso de los usuarios, y de los diferentes aspectos de la aplicación relacionados con el acceso, control y seguridad de los servidores.

Gestión de datos.

Subsistema encargado de todo lo relacionado con la introducción y salida de los datos. Formatos, estructura y características de los mismos. Así como la posibilidad de implementación de importaciones masivas de datos Característica ya contemplada en el documento de planificación de este proyecto, pero que s de carácter secundario. También será el encargado de generar los listados de datos según las necesidades especificadas para los mismos por los usuarios.

Estadísticas.

Subsistema creado para poder dar cumplimiento a otro de los objetivos secundarios, que es el de realizar ciertos cálculos y gráficos estadísticos complejos sobre los listados de datos originales. Debe ser un subsistema aparte debido a que este tipo de gestión será muy concreta y aplicada a este fin. También el hecho de tenerlo totalmente separado, y al ser contemplado como un objetivo secundario a este proyecto, posibilita su desarrollo en un ampliación futura.

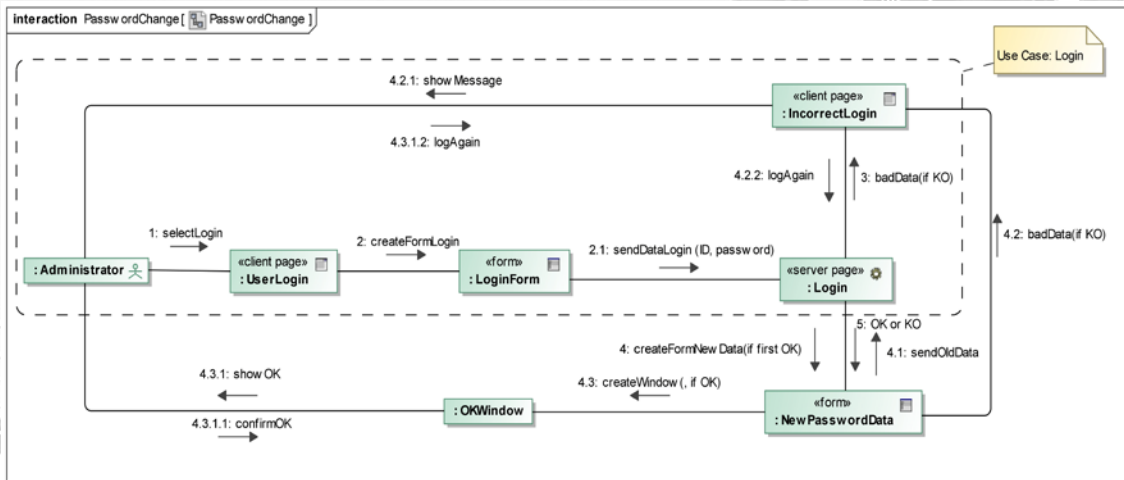
Diseño

DIAGRAMAS DE ACTIVIDAD.

Sólo se realizan para el usuario administrador, ya que este es el que tendrá acceso a todos los apartados de la Web.

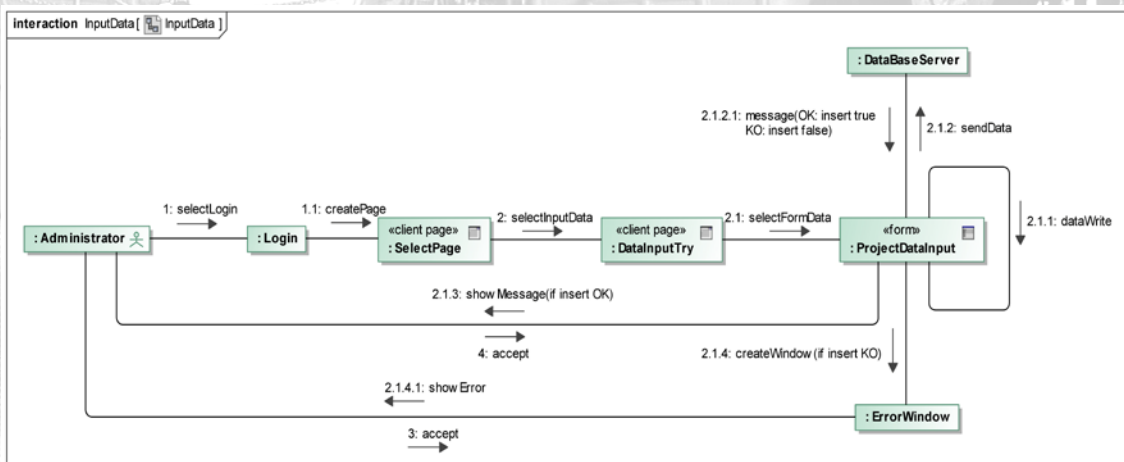
Acceso y cambio de contraseña.

Este diagrama contempla dos casos de uso relacionados del anterior documento de análisis y especificación. La línea punteada enmarca el que es el caso de uso principal, acceso a los apartados de la Web de introducción y edición de datos, el resto sería la parte correspondiente a la edición y cambio de contraseña. Figura 6.



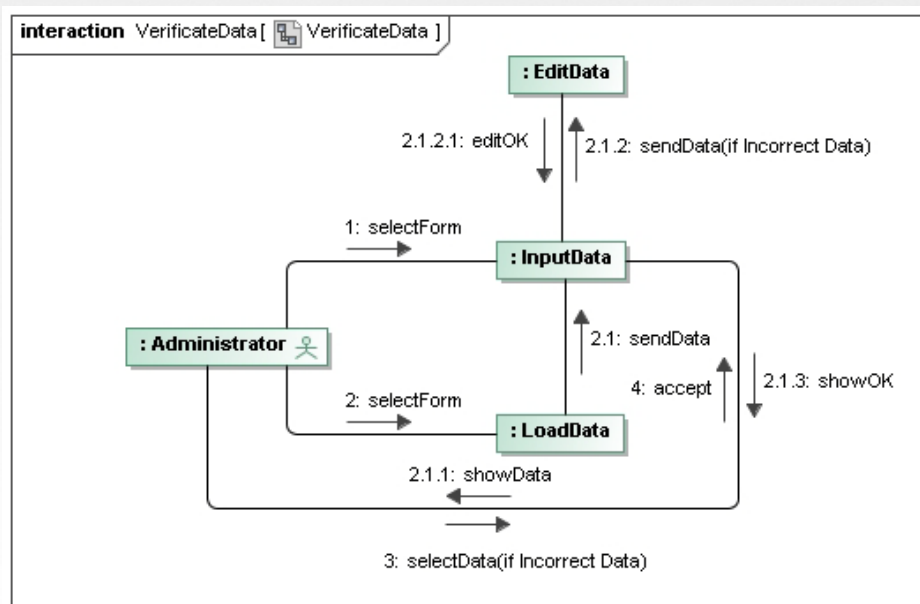
Entrada de datos.

Este es uno de los casos de uso principales, incluye el caso de acceso a la Web (Login) y el nodo físico DataBaseServer donde se almacenan los datos, que en el caso de problemas enviará un mensaje de error que generará una ventana de aviso al usuario con información sobre las posibles soluciones del error. Figura 7.



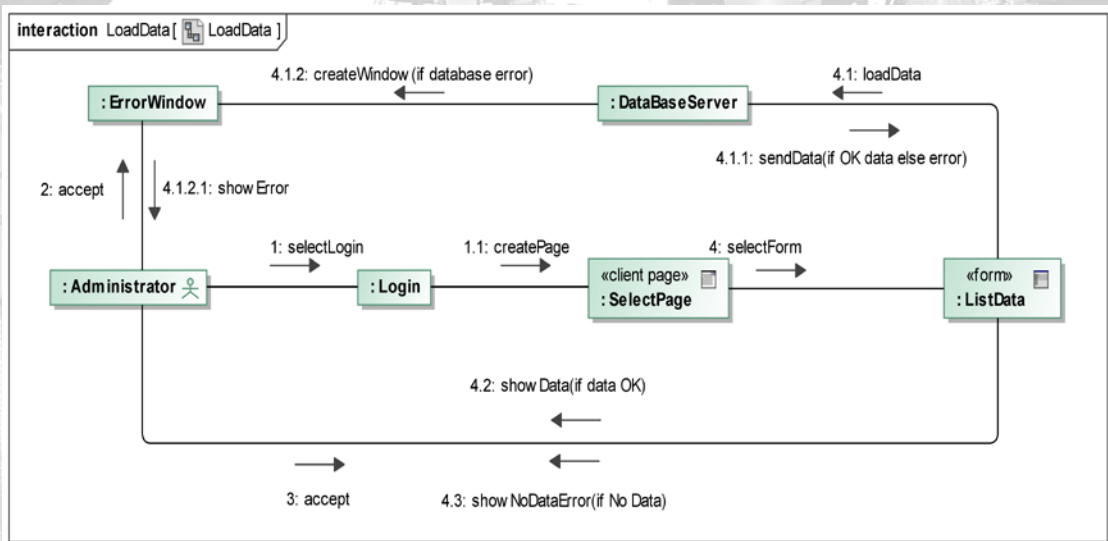
Verificación de datos.

Caso de uso secundario, implica la carga de datos y está altamente relacionado con su edición. Figura 8.



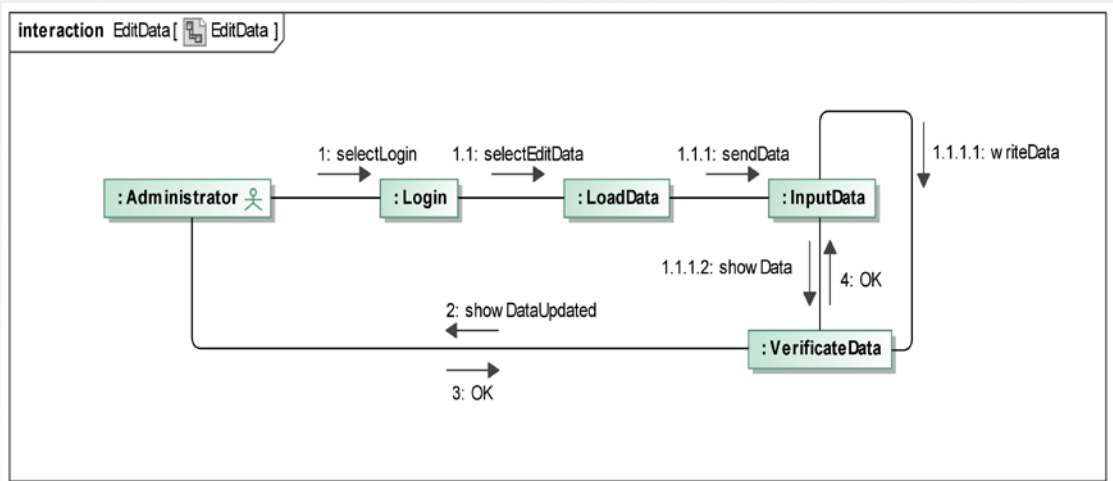
Carga de datos.

Es otro de caso de uso principal, anteriormente mencionado como listado de datos. Usado tanto para la edición de los datos como para generar los listados de los mismos. En el diagrama sólo se muestra el último caso para no complicarlo en exceso, ya que el funcionamiento sería similar, para la carga de datos en edición. Figura 9.



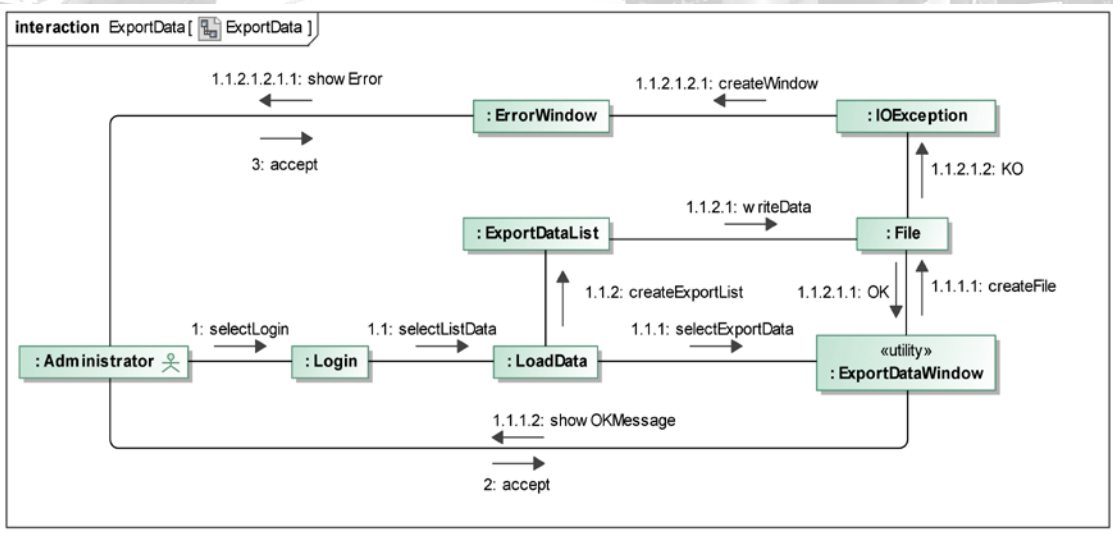
Edición de datos.

Este caso es secundario y está relacionado con el de verificación, tal y como ya se ha mencionado. Permite a diferencia de esta el cambio de aquellos datos que se consideren incorrectos. Se podría haber presentado en el mismo diagrama que la anterior, se ha hecho en dos diferentes por la característica expresada en este apartado. La verificación sería de carácter visual y previo a enviar los datos a la base de datos y la edición supondría que los datos se extraen de la base de datos y permite cambios en los mismos. Figura 10.



Exportar datos.

Caso de uso principal, relacionado con los listados y carga de datos. Se ha priorizado su implementación ya que esta funcionalidad está siendo requerida continuamente. Conlleva el uso de funciones de acceso y escritura al disco duro, y el control e información de los errores que se puedan generar. Figura 11.

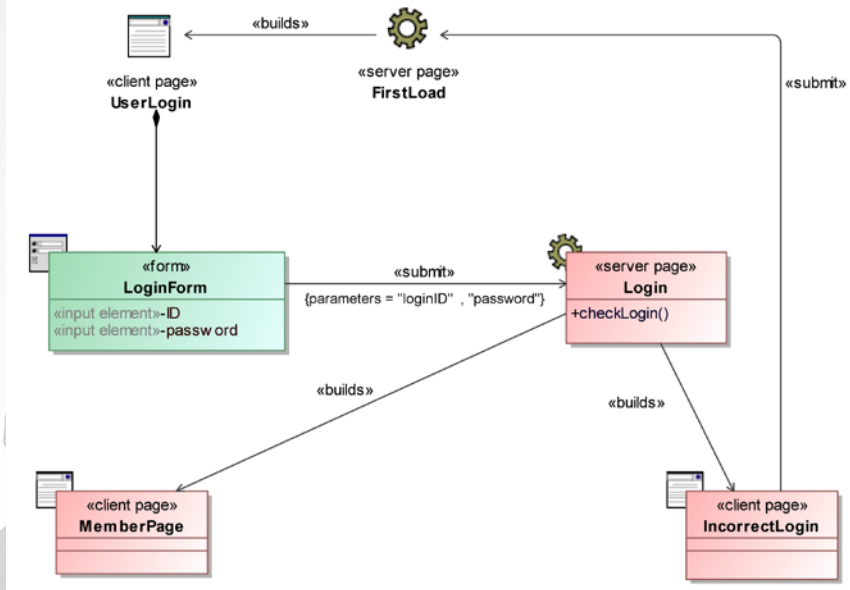


MODELO DE LA WEB.

Modelos del acceso y cambio de contraseña.

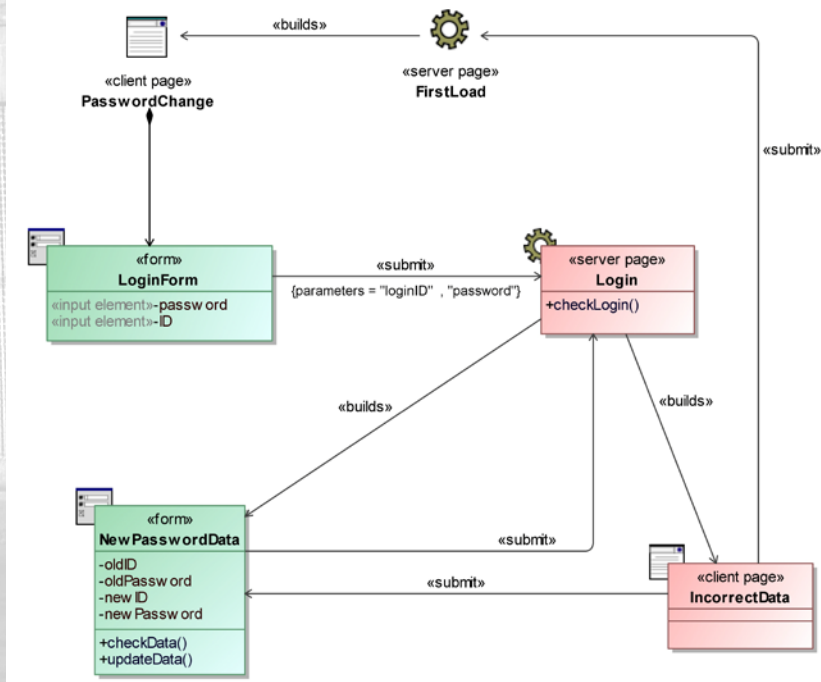
Modelo clásico de acceso mediante un formulario de *login* a la Web. Está basado en los modelos que genera la herramienta de modelado *Magic Draw*⁽²⁴⁾. Contiene los atributos y funciones que se usaran en la implementación. Figura 12.

Login into Prometeo Database Access Web



Modelo clásico de modificación de los datos mediante un formulario de cambio de datos previo *login* en la Web. También está basado en los modelos que genera la herramienta de modelado *Magic Draw*. Además contiene los diferentes atributos y funciones a utilizar en la implementación. Figura 13.

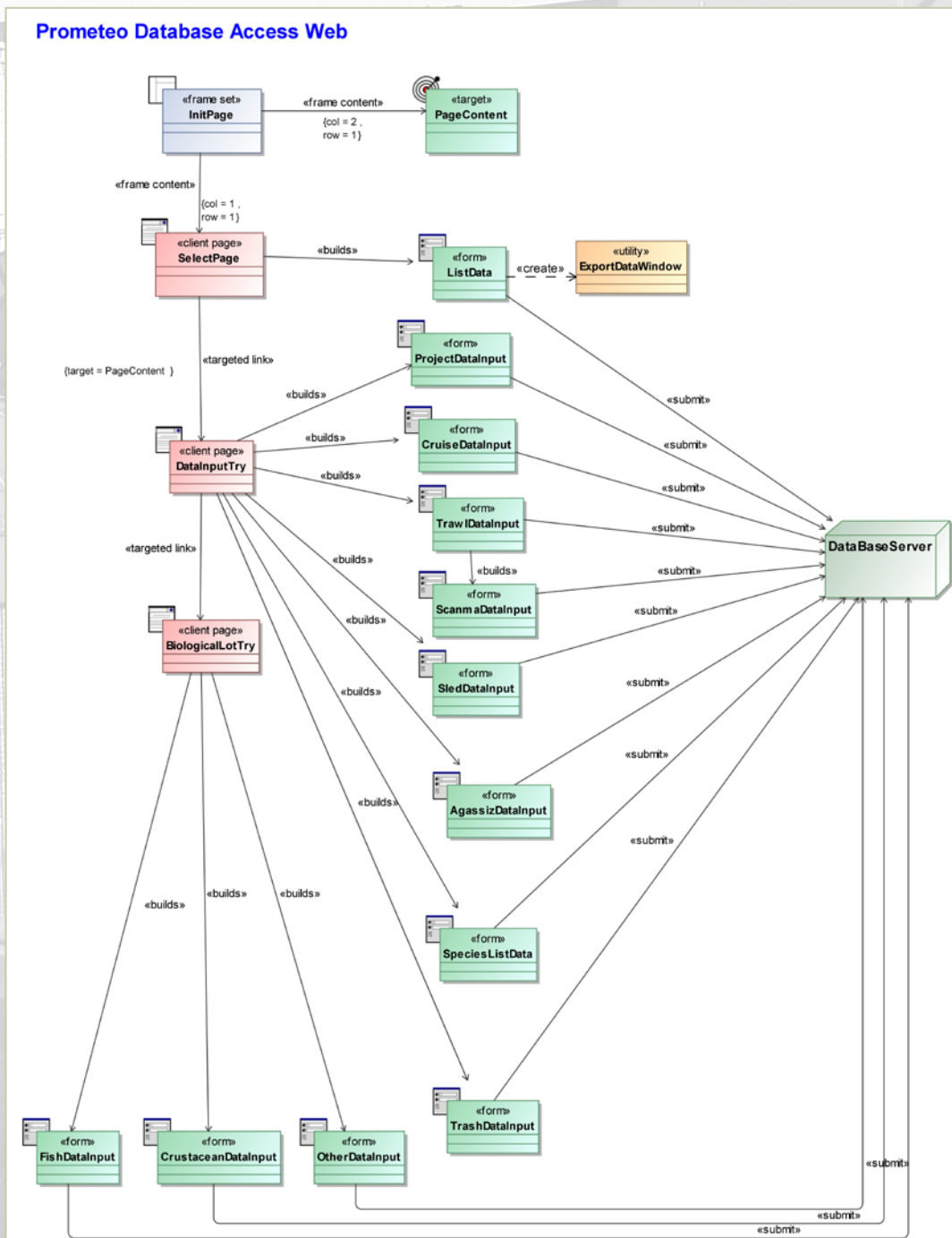
Prometeo Database Access Web: Password Change



Modelo general de la Web.

El modelo sigue la estructura clásica de Web tipo *Book* (de libro). Según las características de acceso del perfil se permitirá el acceso a unos formularios u otros y dentro de ellos tendremos ciertos atributos desactivados, también en función de estos perfiles de acceso. Aquí presento, en principio, todos los necesarios para cubrir los requerimientos esenciales de las especificaciones del proyecto. Los campos y funciones de los formularios los presento en el apartado siguiente en forma de fichas CRC. Pese a que estas están pensadas para detallar clases de programación clásica o también orientada a objetos, nos ayudan a tener un modelo más limpio y claro.

Los formularios de entrada de los datos característicos de cada aparato de muestreo aparecen en la imagen por separado, aunque finalmente se han implementado como un solo nodo, ya que su comportamiento a efectos es este. Cuando se accede a un nodo desde el cliente, se genera un formulario u otro en función del aparato de muestreo, en la máquina del cliente. Enviándose sólo los datos correspondientes a uno de los tipos de formularios que aparecen en la figura. Figura 14.



Fichas CRC.

Formulario	Listada
Descripción	Permite escoger las opciones para generar listados de datos exportables a MS Excel ⁽¹²⁾
Tipos de formulario	Salida de datos.
Características	Accesible a todos los usuarios.
Responsabilidades	Colaboraciones
Carga de los datos según los parámetros seleccionados desde la base de datos.	
Campos	
#project	
#cruise	
#start_date	
#end_date	
#gear	
#sampling_code	
#depth	
#category	
#species	
Funciones	
#export_data	
#load_data	
Formulario	ProjectDataInput
Descripción	Formulario para dar de alta un proyecto de investigación nuevo o corregir y actualizar sus datos.
Tipos de formulario	Entrada o actualización de datos.
Características	Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.
Responsabilidades	Colaboraciones
Introducir todos los datos de un proyecto y darlo de alta en la base de datos. O mediante el código de proyecto (<i>acronym</i>) cargar sus datos y actualizar o corregirlos.	
Campos	
#project_name	
#acronym	
#financial_entity	
#start_date	
#end_date	
#status	
Funciones	
#load_data	
#send_data	
#update_data	
Formulario	CruiseDataInput
Descripción	Formulario para dar de alta una campaña oceanográfica nueva o corregir y actualizar sus datos.
Tipos de formulario	Entrada o actualización de datos.
Características	Accesible para los administradores en modo entrada de datos y actualización. El resto de

	usuarios autorizados sólo entrada.
Responsabilidades	Colaboraciones
Introducir todos los datos de un proyecto y darlo de alta en la base de datos O mediante el código de campaña (<i>name</i>) cargar sus datos y actualizar o corregirlos.	
Campos	
#project	
#ship	
#chief	
#name	
#start_date	
#end_date	
Funciones	
#load_data	
#send_data	
#update_data	
Formulario	TrawlDataInput
Descripción	Formulario para dar de alta un muestro oceanográfico de pesca de arrastre nuevo o corregir y actualizar sus datos.
Tipos de formulario	Entrada o actualización de datos.
Características	Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.
Responsabilidades	Colaboraciones
Introducir todos los datos de un muestreo y darlo de alta en la base de datos O mediante el código de muestreo(<i>trawl_code</i>) cargar sus datos y actualizar o corregirlos.	
Campos	
#cruise	
#trawl_code	
#depth	
#speed	
#date	
#ctd_data	
#stop_winch_time	
#start_winch_time	
#start_winch_latitude	
#end_winch_latitude	
#start_winch_longitude	
#end_winch_longitude	
#wire	
#bottom_on_time	
#bottom_off_time	
start_scanmar_latitude	
#end_scanmar_latitude	
#start_scanmar_longitude	
#end_scanmar_longitude	
#remarks	
Funciones	
#load_data	
#send_data	
#update_data	

Formulario	ScanmarDataInput
Descripción	Formulario para dar de alta los datos de los sensores <i>Scanmar</i> en un muestro oceanográfico de pesca de arrastre nuevo o corregir y actualizar sus datos.
Tipos de formulario	Entrada o actualización de datos.
Características	Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.
Responsabilidades	Colaboraciones
Introducir todos los datos de los sensores <i>Scanmar</i> y darlos de alta en la base de datos O mediante el código de muestreo desde el formulario <i>TrawlDataInput</i> (<i>trawl_code</i>) cargar sus datos y actualizar o corregirlos.	<i>TrawlDataInput</i>
Campos	
#time #height #distance #depth	
Funciones	
#load_data #send_data #update_data	

Formulario	AgassizDataInput
Descripción	Formulario para dar de alta un muestro oceanográfico de patín de arrastre <i>Agassiz</i> nuevo o corregir y actualizar sus datos.
Tipos de formulario	Entrada o actualización de datos.
Características	Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.
Responsabilidades	Colaboraciones
Introducir todos los datos del muestreo y darlos de alta en la base de datos O mediante el código de muestreo (<i>trawl_code</i>) cargar sus datos y actualizar o corregirlos.	
Campos	
#cruise #trawl_code #depth speed #date #ctd_data #stop_winch_time #start_winch_time #start_winch_latitude #end_winch_latitude #start_winch_longitude #end_winch_longitude #wire #flowmeter_1 #flowmeter_2 #remarks	
Funciones	
#load_data #send_data	

#update_data

Formulario

Descripción

Tipos de formulario

Características

Responsabilidades

Introducir todos los datos del muestreo y darlos de alta en la base de datos O mediante el código de muestreo (*trawl_code*) cargar sus datos y actualizar o corregirlos.

Campos

#cruise
#trawl_code
#depth
speed
#date
#ctd_data
#stop_winch_time
#start_winch_time
#start_winch_latitude
#end_winch_latitude
#start_winch_longitude
#end_winch_longitude
#wire
#remarks

Funciones

#load_data
#send_data
#update_data

SledDataInput

Formulario para dar de alta un muestro oceanográfico de patín de arrastre suprabentónico (*Epibentic Sled*) nuevo o corregir y actualizar sus datos.

Entrada o actualización de datos.

Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.

Colaboraciones

Formulario

Descripción

Tipos de formulario

Características

Responsabilidades

Introducir todos los datos de las especies capturadas y darlos de alta en la base de datos O mediante el código de muestreo, género y especie (*sampling_code*, *genas*, *especies*) cargar sus datos y actualizar o

SpeciesListData

Formulario para dar de alta el listado de especies capturadas en un muestro oceanográfico de cualquier tipo. O corregir y actualizar sus datos.

En caso de que las capturas fuesen realizadas mediante patín suprabentónico, en el formulario aparecerán los campos extra de: *upper*, *middle*, *lower*. Esto es así porque este aparato de muestreo pesca en tres niveles y se han de tratar de manera separada.

Entrada o actualización de datos.

Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.

Colaboraciones

corregirlos.

Campos

#cruise
#sampling_code
#genus
#species
#number
#weight

Funciones

#load_data
#send_data
#update_data

Formulario

Descripción

TrashDataInput

Formulario para dar de alta la basura recogida en un muestro oceanográfico de cualquier tipo. O corregir y actualizar sus datos.

Tipos de formulario

Entrada o actualización de datos.

Características

Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.

Responsabilidades

Introducir todos los datos de la basura recogida y darlos de alta en la base de datos O mediante el código de muestreo (*sampling_code*) cargar sus datos y actualizar o corregirlos.

Colaboraciones

Campos

#cruise
#sampling_code
#tipology
#number
#weight
#remarks

Funciones

#load_data
#send_data
#update_data

Formulario

Descripción

FishDataInput

Formulario para dar de alta un individuo (de la familia de los peces óseos o cartilaginosos) capturado en muestro oceanográfico. O corregir y actualizar sus datos.

Tipos de formulario

Entrada o actualización de datos.

Características

Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.

Responsabilidades

Introducir todos los datos sobre el individuo y darlos de alta en la base de datos O mediante el código de muestreo y el número de individuo (*sampling_code*, *ind_number*) cargar sus datos y actualizar o corregirlos.

Colaboraciones

Campos

#cruise
#sampling_code
#ind_number
#species
#size
#weight
#sex
#rep_stage
#gonadal_weight
#remarks

Funciones

#load_data
#send_data
#update_data

Formulario

CrustaceanDataInput

Descripción

Formulario para dar de alta un individuo (de la familia de los crustáceos) capturado en nuestro oceanográfico. O corregir y actualizar sus datos

Tipos de formulario

Entrada o actualización de datos

Características

Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada

Responsabilidades

Introducir todos los datos sobre el individuo y darlos de alta en la base de datos O mediante el código de muestreo y el número de individuo (*sampling_code, ind_number*) cargar sus datos y actualizar o corregirlos

Colaboraciones

Campos

#cruise
#sampling_code
#ind_number
#species
#size
#weight
#sex
#rep_stage
#parasite
#remarks

Funciones

#load_data
#send_data
#update_data

Formulario

OtherDataInput

Descripción

Formulario para dar de alta un individuo (de cualquier familia de animal (no peces y no crustáceos) capturado en nuestro oceanográfico. O corregir y actualizar sus datos.

Tipos de formulario

Entrada o actualización de datos.

Características

Accesible para los administradores en modo entrada de datos y actualización. El resto de usuarios autorizados sólo entrada.

Responsabilidades

Colaboraciones

Introducir todos los datos sobre el individuo y darlos de alta en la base de datos O mediante el código de muestreo y el número de individuo (*sampling_code*, *ind_number*) cargar sus datos y actualizar o corregirlos

Campos

#cruise
#sampling_code
#ind_number
#species
#size
#weight
#sex
#rep_stage
#remarks

Funciones

#load_data
#send_data
#update_data

MODELO DE DATOS IMPLEMENTADO.

El modelo que aparece en la figura posterior está ya implementado en *MySQL*⁽¹⁹⁾ y representa todos los datos manejados en el grupo de investigación: físicos, biológicos y de buques de pesca. Ha sido desarrollado en su totalidad por mi aplicando los conocimientos adquiridos en las asignaturas *Bases de Datos I* y *Bases de Datos II*.

Se han tenido que introducir algunos cambios, sobre el modelo presentado previamente, para reflejar el hecho de que aparatos como el patín suprabentónico o el multicorer realizan capturas simultáneas que deben ser tratadas por separado. En concreto el patín suprabentónico captura en tres niveles de altura, y las capturas de organismos en cada uno de los niveles se deben tratar por separado, cómo si se hubiesen realizado en muestreos diferentes.

Solamente introduciré una explicación de los elementos directamente relacionados con este trabajo, en el que se han aplicado los siguientes razonamientos en función de la forma de trabajar y de las características de los datos. Para las tablas que sufren modificación y actualización tenemos:

- Tabla *project*: recoge los datos sobre un proyecto oceanográfico.
- Tabla *cruise*: recoge los datos de un muestreo oceanográfico, de un proyecto concreto y se realiza con un barco concreto (tabla *boat*).
- Tabla *sampling*: relaciona un muestreo con el áreas geográfica, geomorfológica o de caladero de pesca (tablas *geographical_area*, *geomorphological_area* y *fishing_ground*) donde se desarrolla, así como con la campaña oceanográfica en la que se enmarca (tabla anterior). Finalmente está relacionada con la tabla que contiene todos los muestreos realizados (ver siguiente).
- Tabla *sampling_gear_data*: a partir de esta se aplica el concepto de herencia en bases de datos, de manera disyunta y total. Se relaciona con tres tablas, la que contiene las descripciones de los aparatos de muestreo (*sampling_gear*) y las tablas de muestreo en un punto o en línea (*point_sampler_data* y *line_gear_data*). De estas dos, es la última tabla y sus posteriores relaciones las que nos interesan en este trabajo. La otra línea queda para las posteriores ampliaciones de funcionalidades del trabajo aquí desarrollado.
- Tabla *line_gear_data*: almacena los datos de los aparatos que muestrean mediante arrastre (patines y redes de pesca diversas). De estos existen unos que van a medias aguas (tabla descendiente *middle_trawl_water*) y los ROV que no nos interesan. Y los que muestrean arrastrados por el fondo marino (tabla *bottom_trawl_data*).

- Tabla *botton_trawl_data*: datos de los aparatos de muestreo en fondo (patines y OTMS), de esta emanar las relaciones para el patín suprabentónico (*epibentic sled*) que se trata de manera aparte, y para los otros dos que todavía tiene alguna característica en común (tabla *trawl_gear_data*).
- Tabla *trawl_gear_data*: finalmente en esta tabla se separan el patín *Agassiz* y la red OTMS (tablas *otms* y *agassiz*).
- Tablas *lower*, *middle* y *upper*: almacenan los datos característicos de cada nivel de pesca del patín suprabentónico.
- Tablas *otms*, *scanmar* y *agassiz*: datos característicos de estos aparatos y una parte de los que recogen los sensores *Scanmar*.

Ahora trataremos las capturas, cada tabla con las características diferenciales de cada aparato (*lower*, *middle*, *upper*, *agassiz* y *otms*) está relacionada con las capturas que estos realizan: tabla *catch* o *species_list*. Según sea un aparato de muestro u otro.

- Tabla *catch*: datos de la capturas realizadas por los aparatos *Agassiz* y *OTMS*.
- Tabla *species_list*: los datos de las diferentes especies capturadas en un muestreo de arrastre de fondo, sea con el aparato que sea. Esta relacionada con la tabla anterior para *OTMS* y *Agassiz*.
- Tabla *individual data*: relacionada con la tabla anterior, para conocer el muestreo en que se ha capturado el individuo. Contendrá los datos comunes a todos los individuos capturados por la técnica que sea. Se especializará mediante herencia en disyunta y total en las tablas *fish* y *crustacean* que almacenarán los datos característicos a estas familias.

Finalmente la enumeración de tablas siguientes es necesaria en este trabajo, pero se modifican raramente, ya que sus datos son bastante fijos:

- *fishing_ground* (lista de caladeros de Cataluña).
- *geographical_area* (lista de áreas geográficas del Mediterráneo).
- *geomorphological_area* (división geomorfológica de los fondos marinos).
- *sampling_gear* (lista de aparatos de muestreo y sus descripciones).
- *sex* (macho, hembra, indeterminado, no definido o juvenil).
- *category* (lista de categorías animales).
- *species* (lista de especies marinas).
- *trash_tipology* (lista de los diferentes tipos de basura aparecidos en las pescas).

Figura 15.

Implementación

La fase de implementación es la que tiene un grado de criticidad, complejidad y posibilidad de retraso mayores, cómo finalmente así ha sucedido, hallándose en estos momentos en un 50 % de cumplimiento aproximado. A medida que he avanzado en la implementación de las funcionalidades descritas en fases previas, he observado la necesidad de incluir elementos dinámicos en los formularios del producto para tener de este modo la funcionalidad deseada. Ello me ha llevado a introducirme en el aprendizaje y comprensión del lenguaje de programación *JavaScript* que se ejecuta del lado del cliente, y de las tecnologías que lo utilizan para ofrecer diferentes tipos de soluciones de Web dinámicas: Ajax y AHAH. Habiéndome decantado finalmente por la última, ya que en *Drupal*⁽²⁰⁾ ofrece una gran facilidad para el uso de *JavaScript* sin tener que ser un programador experto del mismo.

TECNOLOGÍAS UTILIZADAS

Hay en la Red paquetes de productos destinados a instalar y gestionar servidores Web (*WampServer*, *AppServ*, etc.), que incluyen lo que a continuación detallo en un conjunto. Sin embargo he optado por una instalación por separado de cada uno de los componentes y tecnologías, para así aprovechar la instalación de la base de datos que ya se tenía realizada.

Para este proyecto se ha utilizado:

Máquina virtual: *VMware Player* v. 3.0.1

Servidor: Linux Ubuntu 10.04 LTS

Gestor de bases de datos: MySQL⁽¹⁹⁾ v. 5.1.41

Servidor Web: Apache⁽²⁵⁾ v. 2.2.14

Idioma de programación principal: PHP⁽²⁷⁾ 5

Contenido dinámico: AHAH basado en JavaScript

Sistema gestor de contenidos: *Drupal*⁽²⁰⁾ v. 6.22

ENTORNO DE PROGRAMACIÓN

Como entorno de programación principal se ha utilizado el IDE (*Integrated Development Environment*) **NetBeans**⁽²⁶⁾ versión 6.9.1. Se caracteriza por ser multiplataforma, multilenguaje de programación y tener una interfaz amigable e innumerables *plugins* que favorecen la programación de aplicaciones.

GESTOR DE CONTENIDO

Se ha decidido utilizar un sistema gestor de contenidos (Content System Management – CMS), que facilitará la labor de gestión y configuración de nuestra Web. En nuestro caso el gestor elegido ha sido *Drupal*⁽²⁰⁾, ya mencionado con anterioridad. Esta herramienta tiene una gran implantación en la Red, está avalada por miles de desarrolladores y se utiliza en más de medio millón de Webs en todo el mundo. Como ejemplos destacables tenemos a la [Casa Blanca des los Estados Unidos de América](#), [TheEconomist](#), [Examiner](#) o inclusive páginas Web de renombrados desarrolladores de software como [Symantec](#). Esto nos da garantías de que estamos tratando con una herramienta segura y robusta, ya que estas Webs tienen gran número de accesos diarios, y por su renombre deben implementar un alto grado de seguridad y robustez.

Drupal⁽²⁰⁾ será el núcleo de la aplicación Web para trabajo colaborativo diseñada para la gestión de los datos del grupo de investigación. Nos permitirá en primer término la gestión segura y transparente de los diferentes perfiles de usuario y las características multidioma que se incorporarán en un futuro.. También en su caso contenido estático de tipo genérico como páginas de presentación, noticias, etc. a las cuales se puede permitir un acceso universal, pero esto no es el objeto de este trabajo.

Por otro lado *Drupal*⁽²⁰⁾ posee un gran número de funcionalidades ya desarrolladas como un API (*Application Programming Interface*) de creación de formularios Web, utiliza de forma nativa la base



de datos *Open Source MySQL*⁽¹⁹⁾ que, cómo ya sabemos, es el almacén de datos elegido para este trabajo.

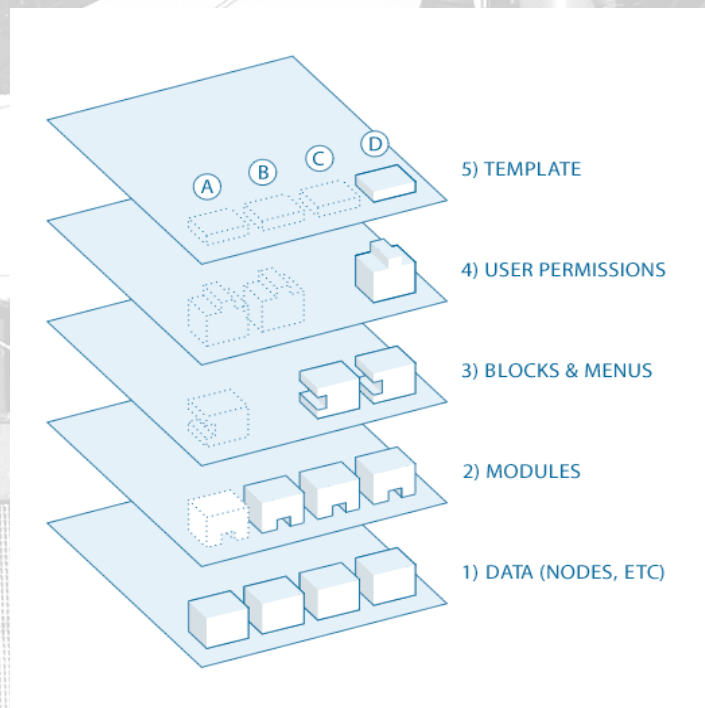
Por último *Drupal*⁽²⁰⁾ es un software diseñado en y para funcionar con el lenguaje de programación PHP⁽²⁷⁾. El cual ha sido elegido para el desarrollo de nuestra Web como idioma prioritario.

Características del gestor

Drupal⁽²⁰⁾ es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto (Open Source), con licencia GNU/GPL, escrito en PHP⁽²⁷⁾, desarrollado y mantenido por una cativa comunidad de usuarios. Destaca por la calidad en su código y de las páginas Web generadas respetando los estándares de la Red, por último enfatizar la usabilidad y consistencia de todo el conjunto.

El diseño de *Drupal*⁽²⁰⁾ es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, por su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales que tiene, hace que pueda ser adecuado para realizar muchos tipos diferentes de sitios Web (Sites).

Esquema del flujo de programación de una aplicación Web realizada en *Drupal*⁽²⁰⁾. Figura 16.



Características principales:

- Sistema de ayuda online para usuarios y administradores.
- Contenido totalmente indexado en tiempo real facilitando la búsqueda.
- Código abierto (*Open Source*), posibilidades de ampliación o adaptación según nuestras necesidades.
- Módulos que proporcionan un sinfín de funcionalidades.
- *Personalización* del contenido y presentación según las preferencias de los usuarios.
- Direcciones Web (*URL*) *amigables*
- Autenticación de usuarios.
- Permisos basados en roles.
- Control de versiones.
- Enlaces permanentes (*Permalinks*).
- Objetos de Contenido (*Nodos*) y *Plantillas* (*Templates*).

- Sindicación del contenido.
- Herramienta para agregar noticias.
- Soporte de *Blogger API*.
- Independencia de la base de datos, permite incorporar fácilmente soporte para otras bases de datos.
- Multiplataforma.
- Múltiples idiomas y localizaciones.
- Administración vía Web.
- Análisis, seguimiento y estadísticas de uso del Web.
- Registros e informes de toda la actividad y los sucesos.
- Comentarios enlazados.
- Encuestas on-line configurables.
- Foros de discusión.
- Libre y colaborativo.
- Control de congestión para habilitar y deshabilitar determinados módulos o blogs.
- Sistema de *caché* para eliminar consultas a la base de datos.

Módulos instalados

A continuación tenemos el listado de los módulos *Drupal*⁽²⁰⁾ instalados para el funcionamiento de nuestra aplicación, así como para realizar diferentes pruebas de adecuación de los mismos a las funciones de nuestra Web.

De todo el listado, los módulos imprescindibles para el desarrollo y adecuado funcionamiento de nuestra aplicación (especialmente los pertenecientes a los cuatro últimos grupos), son los correspondientes a: CCK, Core required, Fecha/Hora, Development, PROMETEO, Other y Views.

CCK (Creation Content Kit):

1. AHAH Dependent Fields 6.x-1.0-alpha1 Allows CCK fields to specify a controller field and react to its changes via AHAH. Depends on: Content (enabled)
2. Conditional Fields 6.x-2.0-beta2 Content fields and group's visibility based on the values of user defined 'trigger' fields. Depends on: Content (enabled)
3. Content 6.x-2.9 Permitir a los administradores definir nuevos tipos de contenidos. Required by: Conditional Fields (enabled), Content Copy (enabled), Content Permissions (enabled), Content Taxonomy (enabled), Content Taxonomy Autocomplete (enabled), Content Taxonomy Options (enabled), Content Taxonomy Tree (enabled), Date (enabled), Date Tools (enabled), AHAH Dependent Fields (disabled), Fieldgroup (enabled), Hierarchical Select Field Selector (enabled), Node Reference (enabled), Number (enabled), Option Widgets (enabled), Text (enabled), User Reference (enabled), Hierarchical Select Content Taxonomy (enabled)
4. Content Copy 6.x-2.9 Enables ability to import/export field definitions. Depends on: Content (enabled)
5. Content Permissions 6.x-2.9 Set field-level permissions for CCK fields. Depends on: Content (enabled)
6. Content Taxonomy 6.x-2.x-dev Defines a field type for taxonomy terms. Depends on: Content (enabled), Taxonomy (enabled). Required by: Content Taxonomy Autocomplete (enabled), Content Taxonomy Options (enabled), Content Taxonomy Tree (enabled), Hierarchical Select Content Taxonomy (enabled)
7. Content Taxonomy Autocomplete 6.x-2.x-dev Defines an autocomplete widget type for content_taxonomy. Depends on: Content (enabled), Content Taxonomy (enabled), Taxonomy (enabled)
8. Content Taxonomy Options 6.x-2.x-dev Defines an option widget type for content_taxonomy for selects, radios/checkboxes. Depends on: Content (enabled), Content Taxonomy (enabled), Taxonomy (enabled), Option Widgets (enabled).

9. Content Taxonomy Tree 6.x-2.x-dev Defines a dynamic tree widget for Content Taxonomy. Depends on: Content (enabled), Content Taxonomy (enabled), Taxonomy (enabled), Taxonomy Manager (enabled)
10. Fieldgroup 6.x-2.9 Create display groups for CCK fields. Depends on: Content (enabled)
11. Node Reference 6.x-2.9 Defines a field type for referencing one node from another. Depends on: Content (enabled), Text (enabled), Option Widgets (enabled)
12. Number 6.x-2.9 Defines numeric field types. Depends on: Content (enabled)
13. Option Widgets 6.x-2.9 Defines selection, check box and radio button widgets for text and numeric fields. Depends on: Content (enabled). Required by: Content Taxonomy Options (enabled), Node Reference (enabled), User Reference (enabled)
14. Text 6.x-2.9 Defines simple text field types. Depends on: Content (enabled). Required by: Node Reference (enabled), User Reference (enabled)
15. User Reference 6.x-2.9 Defines a field type for referencing a user from a node. Depends on: Content (enabled), Text (enabled), Option Widgets (enabled)

Core optional:

1. Aggregator 6.22 Aggregates syndicated content (RSS, RDF, and Atom feeds).
2. Blog 6.22 Enables keeping easily and regularly updated user web pages or blogs.
3. Blog API 6.22 Allows users to post content using applications that support XML-RPC blog APIs.
4. Book 6.22 Allows users to structure site pages in a hierarchy or outline.
5. Color 6.22 Allows the user to change the color scheme of certain themes.
6. Comment 6.22 Allows users to comment on and discuss published content. Required by: Forum (enabled), Tracker (enabled).
7. Contact 6.22 Enables the use of both personal and site-wide contact forms.
8. Content translation 6.22 Allows content to be translated into different languages. Depends on: Locale (enabled)
9. Database logging 6.22 Logs and records system events to the database.
10. Forum 6.22 Enables threaded discussions about general topics. Depends on: Taxonomy (enabled), Comment (enabled)
11. Help 6.22 Manages the display of online help.
12. Locale 6.22 Adds language handling functionality and enables the translation of the user interface to languages other than English. Required by: Date Locale (enabled), Content translation (enabled)
13. Menu 6.22 Allows administrators to customize the site navigation menu. Required by: Hierarchical Select Menu (enabled)
14. OpenID 6.22 Allows users to log into your site using OpenID.
15. Path 6.22 Allows users to rename URLs.
16. PHP filter 6.22 Allows embedded PHP code/snippets to be evaluated.
17. Ping 6.22 Alerts other sites when your site has been updated.
18. Poll 6.22 Allows your site to capture votes on different topics in the form of multiple choice questions.
19. Profile 6.22 Supports configurable user profiles.
20. Search 6.22 Enables site-wide keyword searching.
21. Statistics 6.22 Logs access statistics for your site.
22. Syslog 6.22 Logs and records system events to syslog.
23. Taxonomy 6.22 Enables the categorization of content. Required by: Content Taxonomy (enabled), Content Taxonomy Autocomplete (enabled), Content Taxonomy Options (enabled), Content Taxonomy Tree (enabled), Forum (enabled), Hierarchical Select Taxonomy (enabled), Taxonomy Manager (enabled), Hierarchical Select Content Taxonomy (enabled), Hierarchical Select Taxonomy Views (enabled)
24. Throttle 6.22 Handles the auto-throttling mechanism, to control site congestion.
25. Tracker 6.22 Enables tracking of recent posts for users. Depends on: Comment (enabled)
26. Trigger 6.22 Enables actions to be fired on certain system events, such as when new content is created.

- 27. Update status 6.22 Checks the status of available updates for Drupal and your installed modules and themes.
- 28. Upload 6.22 Allows users to upload and attach files to content.

Core required:

- 1. Block 6.22 Controls the boxes that are displayed around the main content.
- 2. Filter 6.22 Handles the filtering of content in preparation for display.
- 3. Node 6.22 Allows content to be submitted to the site and displayed on pages.
- 4. System 6.22 Handles general site configuration for administrators.
- 5. User 6.22 Manages the user registration and login system.

Fecha/Hora:

- 1. Date 6.x-2.7 Define campos CCK y controles (widgets) de fecha/hora. Depends on: Content (enabled), Date API (enabled), Date Timezone (enabled). Required by: Date Tools (enabled)
- 2. Date API 6.x-2.7 API de fechas que puede ser usada por otros módulos. Required by: Date (enabled), Date Locale (enabled), Date PHP4 (enabled), Date Popup (enabled), Date Repeat API (enabled), Date Timezone (enabled), Date Tools (enabled)
- 3. Date Locale 6.x-2.7 Allows the site admin to configure multiple formats for date/time display to tailor dates for a specific locale or audience. Depends on: Date API (enabled), Locale (enabled)
- 4. Date PHP4 6.x-2.7 Emular las funciones de PHP 5.2 en PHP4.x, PHP 5.0 y PHP 5.1. Requerido cuando se usa Date API con versiones de PHP inferiores a PHP 5.2. Depends on: Date API (enabled)
- 5. Date Popup 6.x-2.7 Habilita calendarios emergentes y widgets de ingreso y selección de fechas y horas basados en jquery. Depends on: Date API (enabled), Date Timezone (enabled)
- 6. Date Repeat API 6.x-2.7 Date Repeat API para calcular fechas y horas repetidas a partir de reglas iCal. Depends on: Date API (enabled)
- 7. Date Timezone 6.x-2.7 Requerido cuando se usa Date API. Modifica la gestión de zona horaria del sitio y del usuario para usar nombres de zonas horarias en vez de un desplazamiento. Depends on: Date API (enabled). Required by: Date (enabled), Date Popup (enabled), Date Tools (enabled)
- 8. Date Tools 6.x-2.7 Herramientas para importar y autocrear fechas y calendarios. Depends on: Content (enabled), Date (enabled), Date API (enabled), Date Timezone (enabled)

Development:

- 1. Form API Validation 6.x-1.x-dev Enable rules and filters on fapi.

Form Elements:

- 1. Hierarchical Select 6.x-3.7 Simplifies the selection of one or multiple items in a hierarchical tree. Required by: Hierarchical Select Content Taxonomy (enabled), Hierarchical Select Field Selector (enabled), Hierarchical Select Flat List (enabled), Hierarchical Select Menu (enabled), Hierarchical Select Small Hierarchy (enabled), Hierarchical Select Taxonomy (enabled), Hierarchical Select Taxonomy Views (enabled)
- 2. Hierarchical Select Content Taxonomy 6.x-3.7 Use Hierarchical Select as the widget for Content Taxonomy CCK fields. Depends on: Hierarchical Select (enabled), Content Taxonomy (enabled), Hierarchical Select Taxonomy (enabled), Content (enabled), Taxonomy (enabled)
- 3. Hierarchical Select Field Selector 6.x-1.0 Form element to select content types and their fields. Depends on: Hierarchical Select (enabled), Content (enabled)
- 4. Hierarchical Select Flat List 6.x-3.7 Allows Hierarchical Select's dropdown to be used for selecting multiple items in a flat list of options. Depends on: Hierarchical Select (enabled)
- 5. Hierarchical Select Menu 6.x-3.7 Use Hierarchical Select for menu parent selection. Depends on: Hierarchical Select (enabled), Menu (enabled)
- 6. Hierarchical Select Small Hierarchy 6.x-3.7 Allows Hierarchical Select to be used for a hardcoded hierarchy. When it becomes too slow, you should move the hierarchy into the database and write a proper implementation. Depends on: Hierarchical Select (enabled)

7. Hierarchical Select Taxonomy 6.x-3.7 Use Hierarchical Select for Taxonomy. Depends on: Hierarchical Select (enabled), Taxonomy (enabled). Required by: Hierarchical Select Content Taxonomy (enabled), Hierarchical Select Taxonomy Views (enabled)
8. Hierarchical Select Taxonomy Views 6.x-3.7 Use Hierarchical Select for Taxonomy exposed filters in Views. Depends on: Hierarchical Select (enabled), Hierarchical Select Taxonomy (enabled), Views (enabled), Taxonomy (enabled)

Other:

1. AHAH helper 6.x-2.1 The AHAH helper module, making gracefully degradable AHAH-driven forms a no-brainer.
2. AHAH helper demo 6.x-2.1 A demo form the AHAH helper module.
3. Form builder 6.x-1.x-dev Form building framework. Depends on: jQuery UI (enabled), Options element (enabled). Required by: Form builder examples (enabled), Form builder Node UI (enabled), Form builder Webform UI (enabled)
4. Form builder examples 6.x-1.x-dev Form builder support for CCK, Webform, and Profile modules. Depends on: Form builder (enabled), jQuery UI (enabled), Options element (enabled)
5. Form builder Node UI 6.x-1.x-dev Form builder enhancements for node module. Depends on: Form builder (enabled), jQuery UI (enabled), Options element (enabled)
6. Form builder Webform UI 6.x-1.x-dev Form builder integration for the Webform module. Depends on: Form builder (enabled), Webform (enabled), jQuery UI (enabled), Options element (enabled)
7. Node Privacy By Role 6.x-1.6 Provides node level access control based on role membership.
8. Options element 6.x-1.4 A custom form element for entering the options in select lists, radios, or checkboxes. Required by: Form builder (enabled), Form builder examples (enabled), Form builder Node UI (enabled), Form builder Webform UI (enabled)
9. Select (or other) 6.x-2.6 Provides a select box form element with additional option 'Other' to give a textfield.
10. Taxonomy Manager 6.x-2.2 Tool for administrating taxonomy terms. Depends on: Taxonomy (enabled). Required by: Content Taxonomy Tree (enabled)

PROMETEO:

1. PROMETO Database tools Several functions to manage PROMETEO database information.
2. Sampling data input form Module for sampling data input into PROMETEO database

User interface:

1. jQuery UI 6.x-1.x-dev Provides the jQuery UI plug-in to other Drupal modules. Required by: Form builder (enabled), Form builder examples (enabled), Form builder Node UI (enabled), Form builder Webform UI (enabled)

Views:

1. Views 6.x-2.12 Create customized list and queries from your database. Required by: Hierarchical Select Taxonomy Views (enabled), Views exporter (enabled), Views Export xls (enabled), Views UI (enabled)
2. Views Export xls 6.x-1.1-beta1 Plugin to export views to XLS. Depends on: Views (enabled)
3. Views exporter 6.x-2.12 Allows exporting multiple views at once. Depends on: Views (enabled)
4. Views UI 6.x-2.12 Administrative interface to views. Without this module, you cannot create or edit your views. Depends on: Views (enabled)

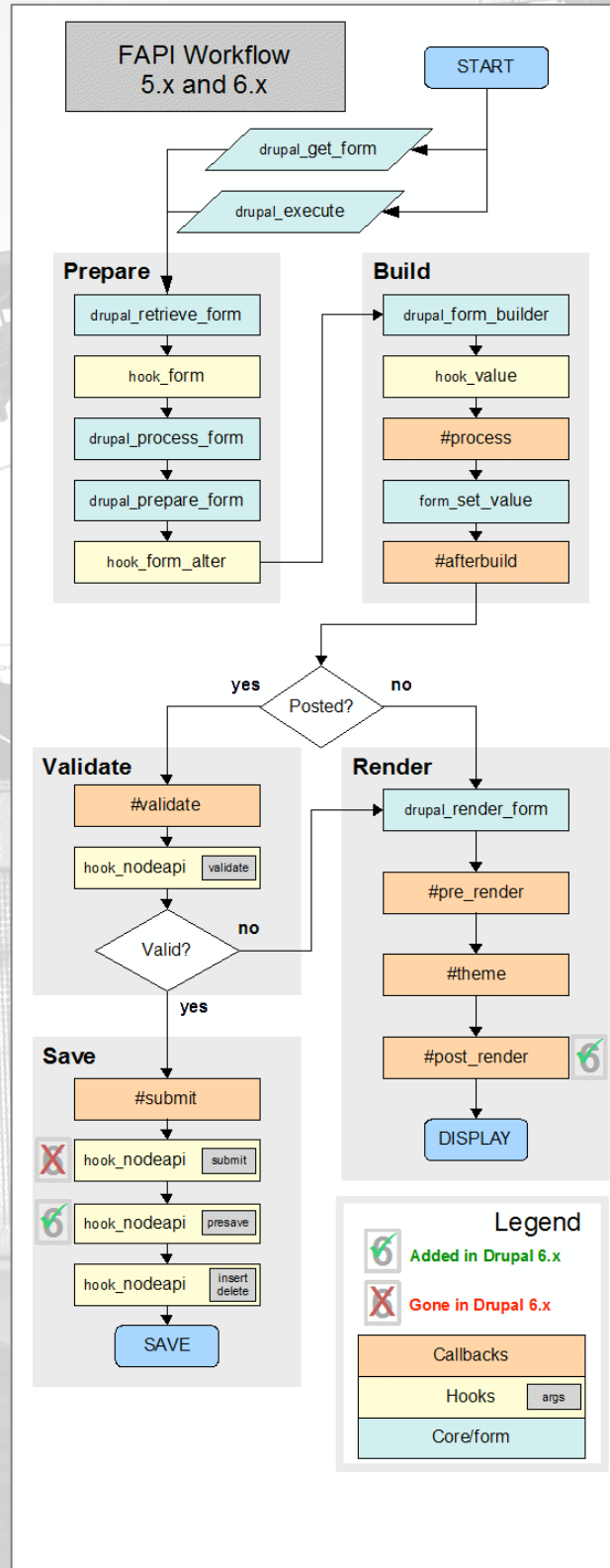
Webform:

1. Webform 6.x-3.11 Enables the creation of forms and questionnaires. Required by: Form builder Webform UI (enabled), Webform Validation (enabled)
2. Webform Validation 6.x-1.4 Add validation rules to webforms. Depends on: Webform (enabled)

MÓDULOS DE PROGRAMACIÓN DESARROLLADOS

Utilizando el entorno de programación, ofrecido por los módulos anteriores, y la capacidad que nos ofrece la API de *Drupal*⁽²⁰⁾ para la construcción de formularios, se han desarrollado los módulos que son el núcleo de nuestra aplicación, principalmente utilizando PHP⁽²⁷⁾ y tecnologías AHAAH.

Para poder comprender mejor el funcionamiento de este entorno, en la figura siguiente (extraída de la Web de soporte de *Drupal*⁽²⁰⁾), se puede ver el flujo de trabajo generado para la creación de un formulario. Figura 17.



Project list

Módulo que únicamente consulta los proyectos de los que hay datos almacenados en la base de datos del grupo de investigación, es la tabla que se nos muestra nada más acceder identificado a la aplicación. La función, de la que sólo menciono su definición, creada a tal fin es:

- *function show_table_project(\$table = null)*: Realiza la consulta a la base de datos y devuelve los proyectos y sus datos almacenados en la misma.

De aquí en adelante seguiré el mismo comportamiento mostrando sólo su definición para todas las funciones de programación del resto de módulos.

Project data

Este módulo permite dar de alta un proyecto dentro de nuestra base de datos. Está operativo por completo, excepto el modo de edición de datos que no está implementado y por tanto tampoco aparece. Tiene las siguientes funciones:

- *function input_data_menu()*: Registra un nodo en nuestra Web, que nos permita llamarlo desde un elemento de nuestro menú.
- *function project_data_form()*: Utiliza la función de *Drupal*⁽²⁰⁾, *drupal_get_form('nuestro formulario')* para retornar el objeto y hacerlos visible.
- *function project_input_form(\$form_state)*: Utilizando el API de construcción de formularios ofrecido por el módulo *Form builder* de *Drupal*⁽²⁰⁾, creamos los campos necesarios para introducir los datos de un proyecto en nuestra base de datos.
- *function project_input_form_submit(\$form_id, &\$form_state)*: Función que sobrescribe el comportamiento por defecto de la función *submit* de *Drupal*⁽²⁰⁾ para realizar la inserción en la base de datos, tal y como nosotros determinemos.
- *function project_input_form_validate(\$form, &\$form_state)*: Utiliza el módulo de *Drupal*⁽²⁰⁾, *Form API Validation* de Pedro Faria, con modificaciones para comprobar los diferentes datos a introducir. Sobrescribe la función *validate* de *Drupal*⁽²⁰⁾.
- *function exists_acronym(\$form_id, &\$form_state)*: Comprobamos que no existe el *acronym* que vamos a insertar en la base de datos. En esta último es clave de tabla pero he preferido hacer la comprobación también en la aplicación.

Cruise Data

Este módulo permite dar de alta una campaña oceanográfica de un proyecto dentro de nuestra base de datos. Está operativo por completo, excepto el modo de edición de datos que no está implementado y por tanto tampoco aparece. Tiene las siguientes funciones:

- *function cruise_input_menu()*. Registra un nodo en nuestra Web, que nos permita llamarlo desde un elemento de nuestro menú.
- *function get_cruise_form()*. Utiliza la función de *Drupal*⁽²⁰⁾, *drupal_get_form('nuestro formulario')* para retornar el objeto y hacerlos visible.
- *function cruise_input_form(\$form_state)*. Utilizando el API de construcción de formularios ofrecido por el módulo *Form builder* de *Drupal*⁽²⁰⁾, creamos los campos necesarios para dar de alta un muestreo de un proyecto en nuestra base de datos.
- *function cruise_input_form_submit(\$form_id, &\$form_state)*. Función que sobrescribe el comportamiento por defecto de la función *submit* de *Drupal*⁽²⁰⁾ para realizar la inserción en la base de datos, tal y como nosotros determinemos.
- *function cruise_input_form_validate(\$form, &\$form_state)*. Utiliza el módulo de *Drupal*⁽²⁰⁾, *Form API Validation* de Pedro Faria, con modificaciones para comprobar los diferentes datos a introducir. Sobrescribe la función *validate* de *Drupal*⁽²⁰⁾.
- *function exists_cruise_id(\$form_id, &\$form_state)*. Comprobamos que no existe el valor de *cruise_id* que vamos a insertar en la base de datos. Aunque es clave de tabla pero prefiero hacer la comprobación previamente en la aplicación.

- *function check_project_date(\$form_id, &\$form_state)*. Comprobamos que las fechas del muestreo que estamos dando de alta se corresponden a fechas dentro del intervalo de tiempo del proyecto en el cual estamos dando de alta ese muestreo.

Sampling Data

Este módulo permite la inserción de los datos de un muestreo, según el aparato que lo haya realizado, dentro de nuestra base de datos. No está operativo por completo, todavía falta implementar las funciones relativas a la comprobación y envío de los datos a la base de datos. Tampoco está implementado el modo de edición de datos. Además introduce en su programación la tecnología *AHAAH* de *Drupal*⁽²⁰⁾, que posibilita el uso de elementos dinámicos dentro de páginas Web, por medio de *JavaScript*.

El uso de *AHAAH* viene determinado para simplificar los nodos de nuestra Web. Simplificando igualmente de este modo, el número de accesos de un usuario a los formularios de introducción de los datos característicos de cada aparato de muestreo en nuestra base de datos. Si recordamos en la fase de diseño se contempla un formulario por cada aparato que se utiliza en una campaña oceanográfica. El concepto se mantiene en el esquema de diseño, pero en la implementación lo he variado, ahora según el aparato de muestreo escogido por el usuario, se mostrarán los campos de datos característicos del mismo. Seguido muestro un fragmento de código, que tiene las líneas que implementan *AHAAH* resaltadas en amarillo:

```

.....
ahah_helper_register($form, $form_state);
.....

$form['sampling_form']['gear'] = array(
  '#weight' => '10',
  '#multiple_toggle' => '1',
  '#type' => 'select',
  '#options' => gear_list(),
  '#title' => t('Gear'),
  '#key_type_toggled' => '1',
  '#default_value' => $usage_default_value,
  '#ahah' => array(
    'event' => 'change',
    // This is the "magical path". Note that the parameter is an
    array of
    // the parents of the form item of the wrapper div!
    'path' => ahah_helper_path(array('sampling_form')),
    'wrapper' => 'sampler-info-wrapper',
  ),
);
$form['sampling_form']['update_gear'] = array(
  '#type' => 'submit',
  '#value' => t('Update gear'),
  // Note that we can simply use the generic submit callback
provided by the
  // ahah_helper module here!
  // All it does, is set $form_state['rebuild'] = TRUE.
  '#submit' => array('ahah_helper_generic_submit'),
  // We set the 'no-js' class, which means this submit button will
be hidden
  // automatically by Drupal if JS is enabled.
  '#attributes' => array('class' => 'no-js'),
);

```

Tiene las siguientes funciones:

- *function sampling_form_menu()*. Registra un nodo en nuestra Web, que nos permita llamarlo desde un elemento de nuestro menú.
- *function sampling_input_form(\$form_state)*. Utilizando el API de construcción de formularios ofrecido por el módulo *Form builder* de *Drupal*⁽²⁰⁾, creamos los campos necesarios para introducir los datos de un muestreo en nuestra base de datos.
- *function sampling_form_validate(\$form, &\$form_state)*. Utiliza el módulo de *Drupal*⁽²⁰⁾, *Form API Validation* de Pedro Faria, con modificaciones para comprobar los diferentes datos a introducir. Sobrescribe la función *validate* de *Drupal*⁽²⁰⁾ (por implementar).
- *function sampling_form_submit(\$form, &\$form_state)*. Función que sobrescribe el comportamiento por defecto de la función *submit* de *Drupal*⁽²⁰⁾ para realizar la inserción en la base de datos, tal y como nosotros determinemos (por implementar).
- *function point_gear_data()*. Controla la inserción de datos para un cierto tipo de aparato de muestreo (por implementar).
- *function otms_gear_data()*. Ídem a la anterior función.
- *function sled_gear_data()*. Ídem a la anterior función.
- *function agassiz_gear_data()*. Ídem a la anterior función.
- *function more_scanmar_data()*. Ídem a la anterior función.

Species list data

Este módulo permite la inserción de los datos de las especies capturadas en un muestreo, dentro de nuestra base de datos. No está operativo por completo, todavía falta implementar las funciones relativas a la comprobación y envío de los datos a la base de datos. Tampoco está implementado el modo de edición de datos. Así mismo falta introducir el uso de *AHAAH*, para hacer que el comportamiento de los campos de selección sea jerárquico, de tal manera que sólo se muestren los datos del proyecto, campaña o muestreos seleccionados.

- *function species_form_menu()*. Registra un nodo en nuestra Web, que nos permita llamarlo desde un elemento de nuestro menú.
- *function species_list_form(\$form_state)*. Utilizando el API de construcción de formularios ofrecido por el módulo *Form builder* de *Drupal*⁽²⁰⁾, creamos los campos necesarios para introducir los datos de un muestreo en nuestra base de datos.
- *function species_list_form_submit(\$form_id, &\$form_state)*. Función que sobrescribe el comportamiento por defecto de la función *submit* de *Drupal*⁽²⁰⁾ para realizar la inserción en la base de datos, tal y como nosotros determinemos.
- *function species_list_form_validate(\$form, &\$form_state)*. Utiliza el módulo de *Drupal*⁽²⁰⁾, *Form API Validation* de Pedro Faria, con modificaciones para comprobar los diferentes datos a introducir. Sobrescribe la función *validate* de *Drupal*⁽²⁰⁾.

Individual data

Este módulo permite la inserción de los datos de los individuos pertenecientes a las especies capturadas en un muestreo, dentro de nuestra base de datos. No está operativo.

- *function individual_form_menu()*. Registra un nodo en nuestra Web, que nos permita llamarlo desde un elemento de nuestro menú.
- *function individual_input_form(\$form_state)*. Utilizando el API de construcción de formularios ofrecido por el módulo *Form builder* de *Drupal*⁽²⁰⁾, creamos los campos necesarios para introducir los datos de un muestreo en nuestra base de datos.
- *function individual_input_form_submit(\$form_id, &\$form_state)*. Función que sobrescribe el comportamiento por defecto de la función *submit* de *Drupal*⁽²⁰⁾ para realizar la inserción en la base de datos, tal y como nosotros determinemos.
- *function individual_input_form_validate(\$form, &\$form_state)*. Utiliza el módulo de *Drupal*⁽²⁰⁾, *Form API Validation* de Pedro Faria, con modificaciones para comprobar los diferentes datos a introducir. Sobrescribe la función *validate* de *Drupal*⁽²⁰⁾.

List data

Este módulo permite la selección de los datos de los proyectos, campañas, muestreos, especies y / o individuos que queremos visualizar y que están almacenados en la base de datos. No está operativo. Utilizará *AHAH* para implementar elementos de selección dinámicos: selección jerárquica. Se construirá la consulta a la base de datos en el momento según los campos seleccionados. Un ejemplo del comportamiento buscado lo encontraremos en el prototipo desarrollado, que simula dos ejemplos de salida de datos según los campos seleccionados. Concretamente en el apartado *Data output*.

- *function list_data_form_menu()*. Registra un nodo en nuestra Web, que nos permita llamarlo desde un elemento de nuestro menú.
- *function list_data_form(\$form_state)*. Utilizando el API de construcción de formularios ofrecido por el módulo *Form builder* de *Drupal*⁽²⁰⁾, creamos los campos necesarios para seleccionar los datos.
- *function list_data_form_submit(\$form_id, &\$form_state)*. Función que sobrescribe el comportamiento por defecto de la función *submit* de *Drupal*⁽²⁰⁾ para realizar la inserción en la base de datos, tal y como nosotros determinemos.
- *function list_data_form_validate(\$form, &\$form_state)*. Utiliza el módulo de *Drupal*⁽²⁰⁾, *Form API Validation* de Pedro Faria, con modificaciones para comprobar los diferentes datos a introducir. Sobrescribe la función *validate* de *Drupal*⁽²⁰⁾.

Tools

Módulo de utilidades que contiene todas las funciones de uso genérico de los módulos anteriores, y relacionadas con esta implementación.

- *function check_table(\$table = null)*. Comprueba la existencia de una tabla en la base de datos.
- *function project_list()*. Llena los desplegados que así lo requieran con la lista de proyectos dentro de la base de datos.
- *function boat_list()*. Llena los desplegados que así lo requieran con la lista de barcos dentro de la base de datos.
- *function cruise_list()*. Llena los desplegados que así lo requieran con la lista de campañas oceanográficas dentro de la base de datos.
- *function geographical_area_list()*. Llena los desplegados que así lo requieran con la lista de áreas geográficas dentro de la base de datos.
- *function geomorphological_area_list()*. Llena los desplegados que así lo requieran con la lista de áreas geomorfológicas que hay dentro de la base de datos.
- *function fishing_ground_list()*. Llena los desplegados que así lo requieran con la lista de caladeros registrados en la base de datos.
- *function gear_list()*. Llena los desplegados que así lo requieran con la lista de aparatos de muestreo dados de alta en la base de datos.
- *function sampling_list()*. Llena los desplegados que así lo requieran con la lista de los muestreos dados de alta en la base de datos.
- *function species_list()*. Llena los desplegados que así lo requieran con la lista de las especies dadas de alta en la base de datos.
- *function buttom_form_clear(\$form, &\$form_state)*. Función para limpiar los campos de los formularios.
- *function fieldset_labels_on_left_table(&\$group, \$keep_fieldset = false, \$weight = 0)*. Función de formato, sitúa las etiquetas de los campos de los formularios a la izquierda de los mismos. Autor: Jumo Manninen de la comunidad de desarrolladores de *Drupal*⁽²⁰⁾.
- *function fieldset_subfieldsets_to_table(&\$group, \$header)*. Función de formato, sitúa los campos de los formularios de forma tabular. Autor: Juho Manninen de la comunidad de desarrolladores de *Drupal*⁽²⁰⁾.

- *function render_titleless(\$el)*. Función de soporte, elimina las cabeceras predeterminadas en los elementos de los formularios. Autor: Juho Manninen de la comunidad de desarrolladores de *Drupal*⁽²⁰⁾.

DESCRIPCIÓN DEL WEB

El producto final ha sufrido variaciones en su estética, pero no en funcionalidades, pese a que estas últimas todavía no se hayan desarrollado por completo.

Comentar también que está activada la característica multidioma del Web, pero al no estar totalmente terminado, no está en funcionamiento al 100% mostrando sólo una parte de la misma traducida.

Estética

El tema final de nuestra Web, sobre el que *Drupal*⁽²⁰⁾ le da su aspecto al producto es *Garland*, con la combinación de colores *Aquamarine*. Todo esto es configurable desde la interface de administración de *Drupal*⁽²⁰⁾. Siendo otra gran ventaja el hecho de permitir la separación del aspecto de la aplicación Web de su funcionalidad, ya no sólo de este gestor de contenido, sino de la gran mayoría de los actuales.

Acceso

El acceso como administrador Web, se hará en la URL que muestro a continuación, con el usuario y clave de acceso que siguen:

URL: <http://norit.cmima.csic.es:5800/drupal/>

User: **user**

Password: **password**

Importante: este perfil no se corresponde al de administrador, mencionado anteriormente de la Web, y que tiene otro tipo de funciones asignadas. Únicamente podrán hacer uso de este último los desarrolladores de la aplicación.

El acceso como usuario de la aplicación, se hará en la URL que muestro, con el usuario y clave de acceso siguientes:

URL: <http://norit.cmima.csic.es:5800/drupal/>

User: **user**

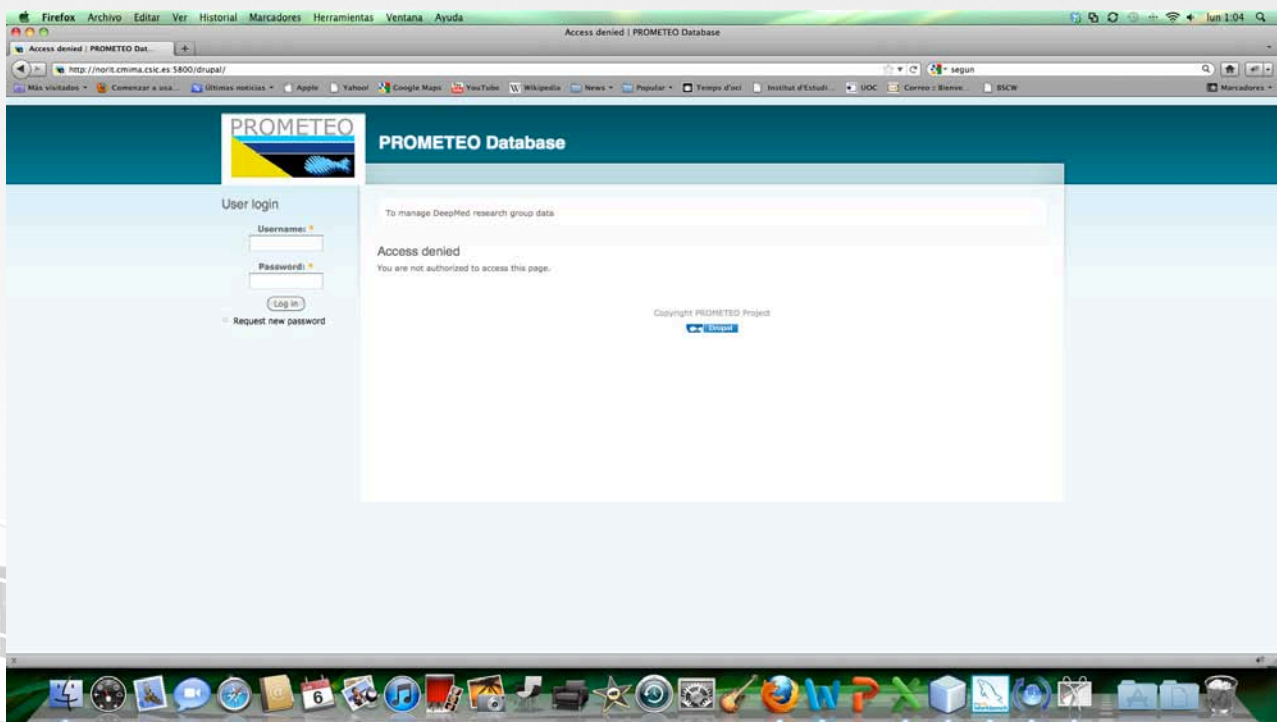
Password: **password**

Pantallas

A continuación mostraré algunas capturas de las pantallas principales en funcionamiento de nuestra aplicación Web.

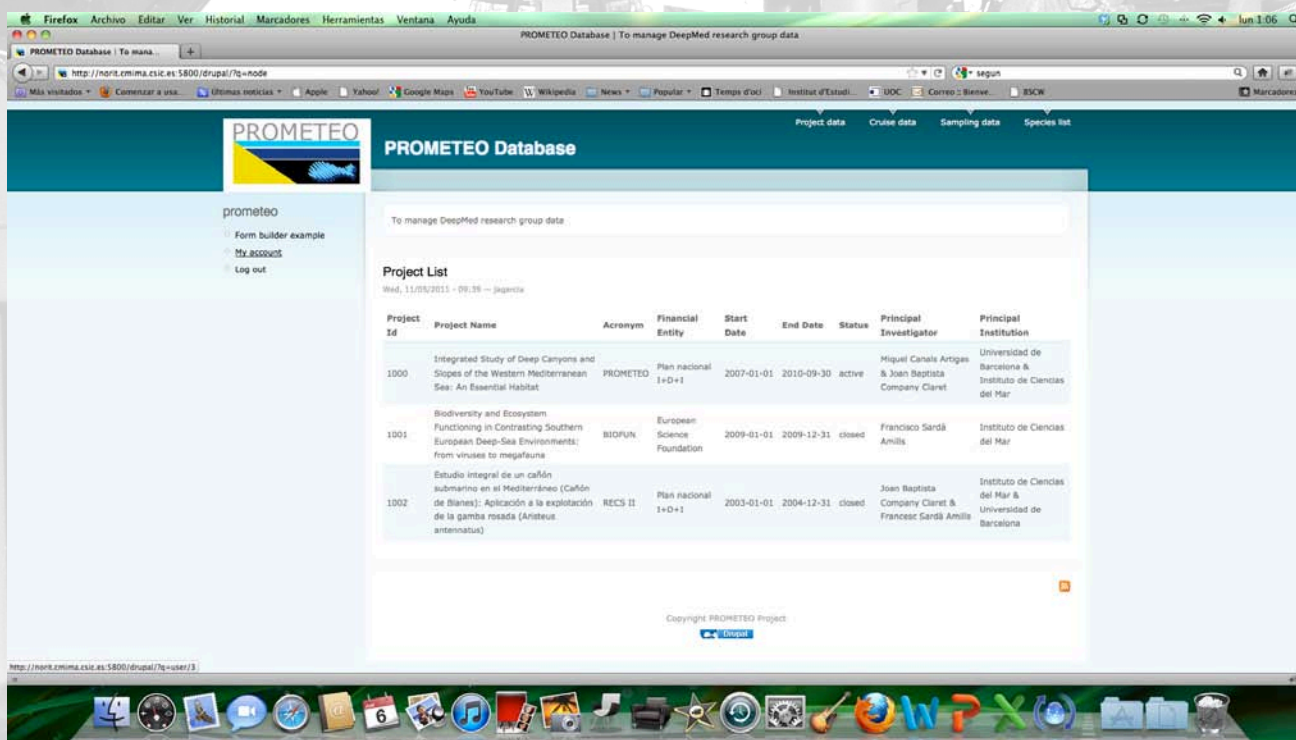
ACCESO

Introducimos las credenciales de usuario, accediendo con un tipo de perfil u otro. Figura 18.



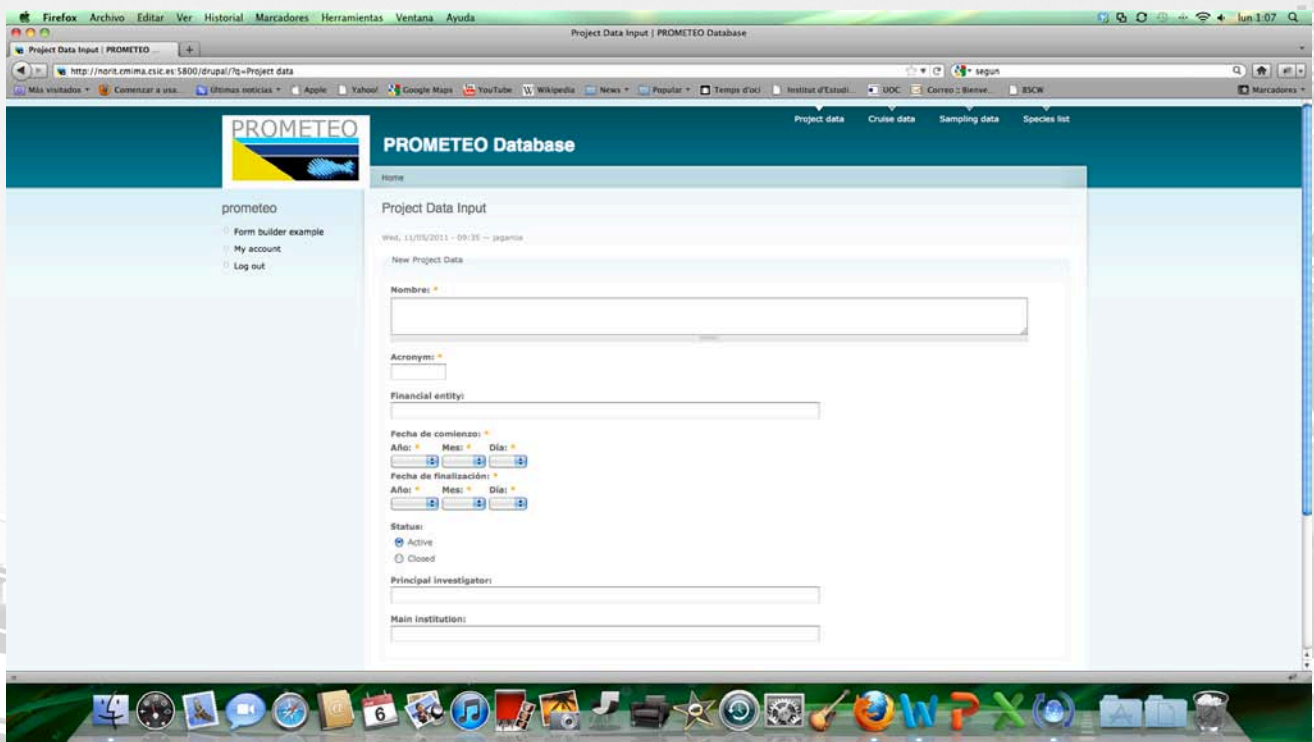
INICIO

Muestra una tabla con los proyectos sobre los que hay datos en la base de datos. En la parte superior aparece el menú con los accesos a los formularios de la Web. Figura 19.



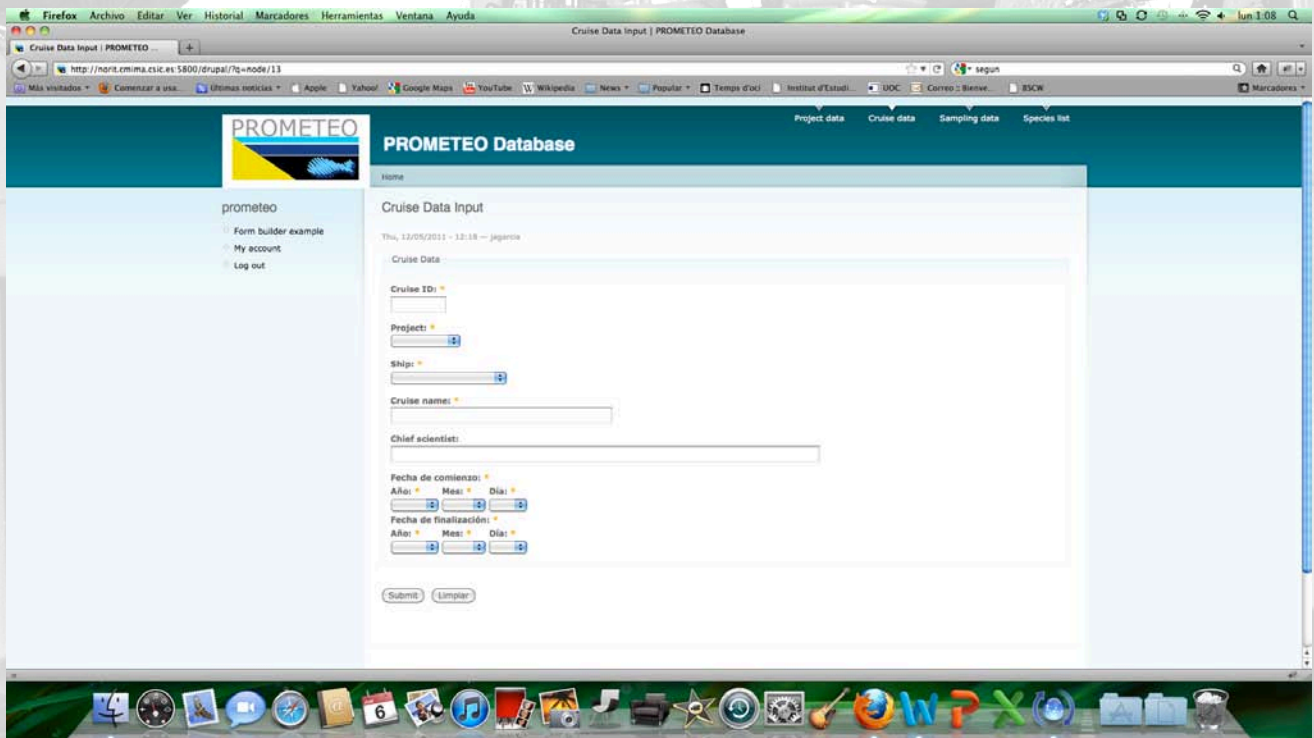
PROJECT DATA

Introducimos los datos en los campos correspondientes y pulsando el botón *submit* serán introducidos en la base de datos si son correctos. Figura 20.



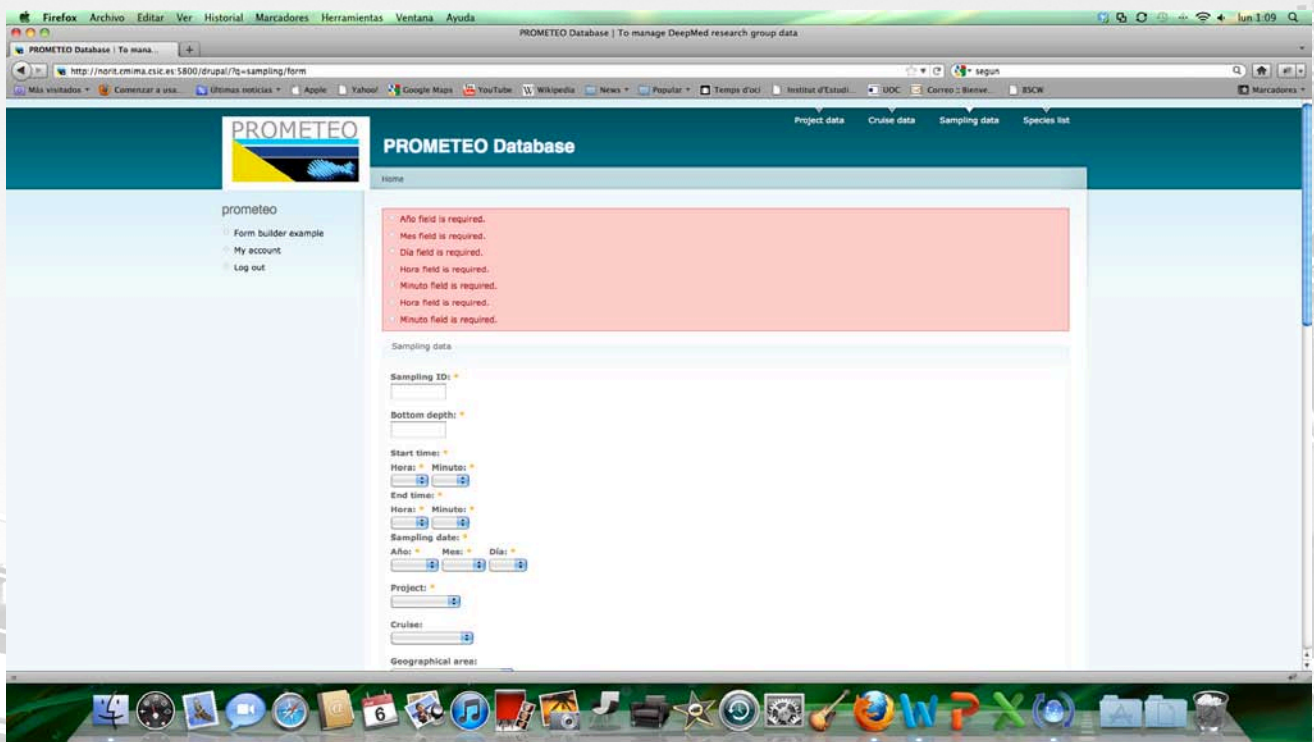
CRUISE DATA

Introducimos los datos en los campos correspondientes y pulsando el botón *submit* serán introducidos en la base de datos si son correctos. Figura 21.



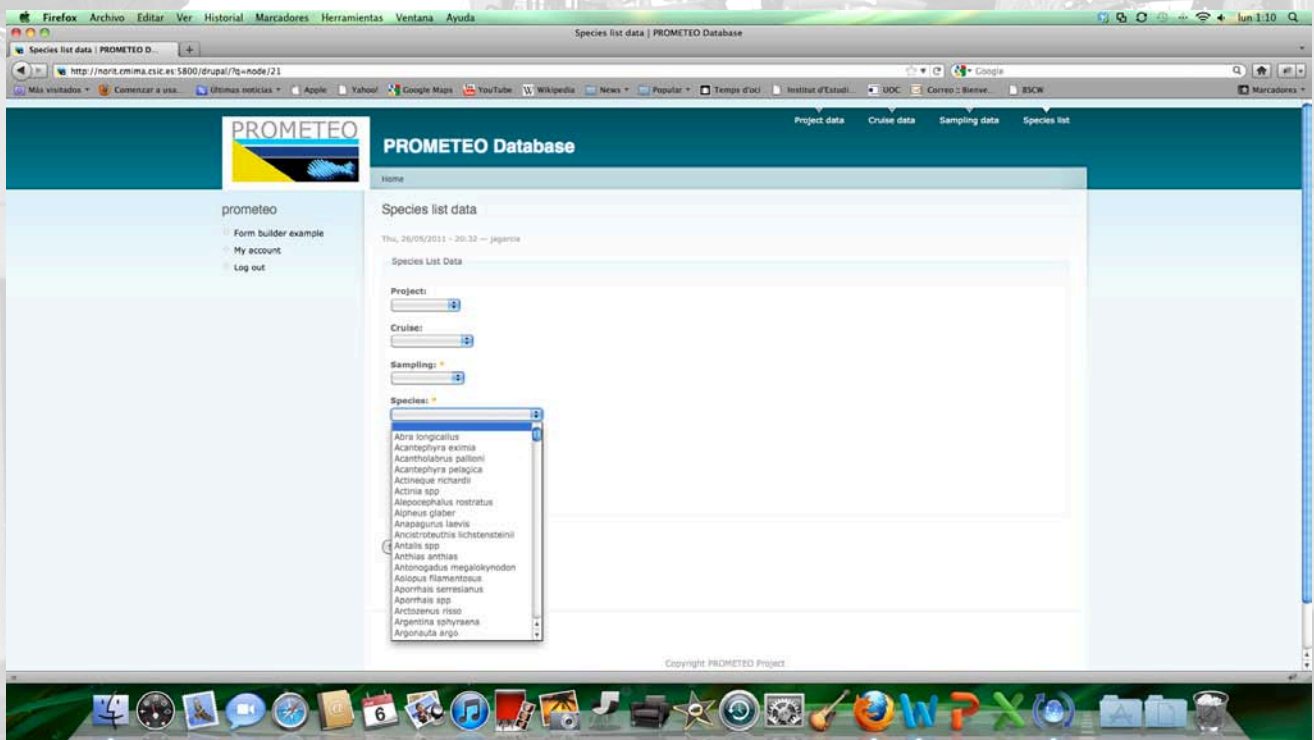
SAMPLING DATA

Sólo esta funcional los desplegables, en el caso del desplegable *gear* (aparece más abajo de lo que esta captura muestra) según el aparato seleccionado, mostrará unos campos u otros. Figura 22.



SPECIES DATA

Sólo están funcionales los desplegables, en la captura se ve el desplegable *species* en funcionamiento. Figura 23.



MEJORAS FUTURAS

Primeramente estará cerrar todo lo relacionado con los objetivos de este trabajo, y que aun está por concluir. La fecha prevista de finalización será el mes de septiembre de este año, y no antes por motivos laborales diversos. Estas tareas que faltan son:

- Terminar los formularios de entrada y salida de datos (prioritaria).
- Introducir la funcionalidad de edición de datos.
- Introducir la funcionalidad de exportar los datos a un fichero.
- Tratamiento de los datos con estadísticas concretas y su representación previa (diciembre 2011).

De cara a un futuro próximo finales del 2011 o primeros del año 2012, se puede añadir contenido público al Web, de carácter general y divulgativo. Será necesario indexarlo en los buscadores para que el público tenga conocimiento y acceso al mismo (en estudio).

Otro aspecto a tener en cuenta, será facilitar el acceso a la comunidad científica y al público en general a los datos, ya sea a todo el contenido de la base de datos o sólo a aquel de carácter más representativo (pendiente de aprobación).

Finalmente se ha pensado poner a disposición del *Coastal Ocean Observatory*⁽²⁸⁾ del Instituto de Ciencias del Mar los datos que genera el grupo. Más concretamente para que puedan ser visualizados en su WebGis (en estudio).

CONCLUSIONES

Este trabajo me ha permitido mejorar mi visión de conjunto de un proyecto de carácter informático. Ya que estaba acostumbrado a una visión de carácter más biológico, habiendo podido aplicar y relacionar, una gran mayoría de los conocimientos adquiridos en el transcurso de los años que llevo en la UOC⁽³³⁾.

Así mismo se han incrementado mis conocimientos en nuevos (para mí) lenguajes de programación (PHP⁽²⁷⁾ y JavaScript) He mejorado sustancialmente la comprensión del modelo cliente / servidor en que está basada Internet, que yo tenía. Además he aprendido el manejo avanzado del gestor de contenidos *Drupal*⁽²⁰⁾, en lo que yo considero han sido uno de los aspectos más críticos.

Otro factor a resaltar en este apartado ha sido el tiempo, pese haber realizado una planificación, que en un principio se pensó correcta y realista, este se ha acortado de forma exponencial impidiendo la total finalización de la aplicación a la que está dirigida este trabajo.

Finalmente con este trabajo se ha buscado una utilidad directa de la aplicación en desarrollo, y que será puesta a prueba en su modo introducción de datos en las futuras campañas oceanográficas que se realizaran en otoño de este mismo año.

GLOSARIO BIOLÓGICO

Agassiz: Aparato de muestreo de macrofauna bentónica, consistente en un armazón cuadrado metálico forrado de malla de red y que se arrastra por la popa de un barco.

Aguas profundas / ecosistema profundo / DEEP-SEA: Ecosistema que ocupa la zona de mayor profundidad de los océanos y mares.

Área de barrido: Magnitud calculada para un red de pesca de arrastre, definida por la coordenada inicial en que toca el fondo y la coordenada final de despegue del fondo y la anchura de la boca de la red. Con ellas se determina un rectángulo en el fondo marino cuya área es sobre la que se ha muestreado.

Área geográfica: Región marina donde transcurre la expedición oceánica.

Área geomorfológica: Referida a la zonación teórica en que se los fondos marinos en función de la profundidad o forma de los mismos. Ej. Plataforma, talud, cañón, etc.

Arte de pesca: Red de pesca.

Bentos: Son los organismos vivos que viven ligados a los fondos marinos. O lo que es lo mismo el fondo del mar es su hábitat.

BIOFUN: Biodiversity and Ecosystem Functioning in Contrasting Southern European Deep-Sea Environments: from viruses to megafauna. Proyecto científico Europeo, financiado por la European Science Foundation, y de la que el grupo de investigación ha sido participante. Año 2009.

Bopírido: Especie animal de isópodo marino parasitaria de muchos crustáceos decápodos marinos.

Captura: Referido al contenido pescado por una red de pesca u otro método de captura.

Categoría: En este entorno referida a la jerarquía superior en la clasificación de las especies marinas dentro de este proyecto.

Cigala (*Nephrops norvegicus*): Especie de crustáceo decápodo marino que vive en el mar Mediterráneo y océano Atlántico. También tiene un gran interés culinario y valor comercial. Se pueden encontrar a más de 600 m. de profundidad, pero su mayor población se haya dentro del mediterráneo en 80 y 400 metros de profundidad, por lo que se cree sean poblaciones diferenciadas⁽²⁸⁾.

Ecosistema: Comunidad de los seres vivos cuyos procesos vitales se relacionan entre sí y se desarrollan en función de los factores físicos de un mismo ambiente.

Epibentic Sled: Traducido como patín suprabentónico, consiste en un armazón metálico con forma de trineo. Tiene tres compartimentos rectangulares dispuestos en altura, a cada uno de los cuales va unida una red terminada en un colector donde se depositan las capturas. Su función es la de capturar la microfauna bentónica

Especie: Jerarquía más baja en la clasificación de las especies vivas y que es de carácter unívoco cuando está unido al género.

Espermatóforo: Órgano del que disponen algunas hembras de especies de crustáceos decápodos marinos, cuya función es la de almacenar el esperma de los machos para fecundar los huevos en un momento posterior.

Flowmeter: Traducido sería “medidor de flujo” (no se usa la traducción), consiste en una especie de hélice espiral fijada a la boca de ciertos aparatos de muestreo (como el “patín”), esta hélice al estar fijada en la boca de la red, gira al ser arrastrada la misma. Conociendo el área de la boca de la red y las vueltas que ha girado este aparato (con una cierta calibración) se puede calcular el volumen de agua filtrado por esa red, y que nos servirá para la normalización de los datos obtenidos mediante ese método de muestreo.

Gamba rosada (*Aristeus antennatus*): Especie de crustáceo decápodo marino que vive en el mar Mediterráneo y costas atlánticas de la península ibérica. Tiene un gran interés culinario y valor comercial. Se han encontrado ejemplares a más de 2000 m. en el mar Mediterráneo, pero su pico de población está entre los 800 y 1000 metros de profundidad.

Género: Jerarquía inmediatamente superior a la de especie en la clasificación de las especies vivas y que es de carácter unívoco unido a la especie. Cuando se clasifica a un ser vivo se habla de género y especie.

Hábitat: Lugar de condiciones apropiadas para que viva un organismo, especie o comunidad animal o vegetal.

HERMINONE: Hotspot Ecosystem Research and Man's Impact on European Sea. Proyecto internacional financiado por el VII programa marco europeo, y en el que el grupo de investigación es participante. Años 2009 al 2011.

Histología: Estudio de los tejidos de organismos vivos. En nuestro caso principalmente se estudian cortes de tejidos gonadales de hembras de especies de peces marinos, para determinar con exactitud su estado de madurez.

Individuo: Referido a cada uno de los ejemplares capturados por un aparato de muestreo.

Índice gonadosomático (GSI): Cálculo realizado en base a la histología y morfometría de un pez.

Isaacs-Kidd Mid-water Trawl (IKMT): Un tipo de red de pesca pelágica de carácter experimental, dedicada a la captura de larvas de peces.

Lote: Parte de una captura que generalmente se corresponde con todos los individuos de una misma especie capturados en un mismo muestreo.

Macrofauna: En este contexto referido a los animales marinos de una determinada talla. Talla intermedia.

Maquinilla: Cabrestante en un barco de pesca generalmente de tamaño considerable.

Meiobentos: Organismos marinos que viven dentro del sedimento. Generalmente son especies de gusanos.

Megafauna: En este contexto referido a los animales marinos de una determinada talla. Talla grande.

Microfauna: En este contexto referido a los animales marinos de una determinada talla. Talla pequeña.

Morfometría: Sería de medidas sobre la forma de un individuo objeto de estudio.

Muestreo oceanográfico / campaña oceanográfica / expedición oceanográfica: Viaje planificado a través de aguas marinas, con objeto de obtención de datos de carácter, físico, químico y biológico. El objetivo es investigar y tener un mayor conocimiento de todos los procesos que influyen en la vida de los seres vivos y que están relacionados con los océanos.

Históricamente se ha considerado como primera expedición oceanográfica, el viaje del navío de la marina británica HMS *Challenger* de 1872 a 1876 ⁽²⁹⁾.

Multicorer: Aparato de muestreo marino en un punto, consistente en un armazón metálico circular sobre el que se montan una serie de tubos plásticos (generalmente transparentes), y que a su vez va montado sobre una estructura de patas en trípode. Se baja mediante la maquinilla al fondo del mar y cuando de posa sobre este un mecanismo de ballesta dispara los tubos plásticos contra el fondo del mar clavándolos en el mismo. Cuando se recoge el cable y el aparato comienza a ascender, la misma ballesta cierra los tubos para que no se pierda lo recolectado. Se puede usar para recolectar el meiobentos aunque se diseñó para extraer sedimento de los fondos marinos.

Núcleo o core: Cada uno de los tubos en que esta compuesto el multicorer.

Otolito: Parte de carácter óseo del órgano sensorial de los peces, y que como característica permite conocer la edad y condiciones ambientales en que se ha desarrollado de un organismo. La forma de hacerlo es muy similar a la que se utiliza en el estudio de los árboles, ya que en los otolitos también hay reflejados anillos de crecimiento.

Otter Trawl Maireta System (OTMS): Red de pesca de arrastre bentónica, utilizada para capturas de megafauna. Es muy similar a la que utilizan las barcas de pesca, diferenciándose principalmente de la de estas en su tamaño y luz de malla. Este sistema esta patentado por los investigadores al que va dirigido este proyecto.

Pesca: Sector de la industria considerado primario junto a la agricultura, que se dedica a la extracción de organismos marinos por medio de diferentes técnicas y cuyo objeto es la alimentación humana.

Peso gonadal: Peso de los órganos de reproducción que para este proyecto principalmente son femeninos, y que nos pueda dar orientación sobre la madurez sexual de un individuo.

PROMETO: Integrated Study of Deep Canyons and Slopes of the Western Mediterranean Sea: An Essential Habitat. Proyecto científico nacional, financiado por el plan nacional I+D+I, y en el que el grupo de investigación ha sido participante. Años 2007 al 2009.

Rangos batimétricos: Referidos a la variación en profundidad dentro de la cuenca marina.

Rectangular Mid-water Trawl (RMT): Un tipo de red de pesca pelágica de carácter experimental, dedicada a la captura de larvas de peces.

Recursos marinos renovables: se consideran recursos (cómo podrían ser los minerales) a los organismos marinos que se obtienen por pesca extractiva o en la acuicultura.

ROV: Remote Operated Vehicle. Es un robot submarino no tripulado y conectado a un barco en la superficie por medio de un cable largo. La energía y los órdenes se envían mediante un mando a distancia a través del cable al ROV (fuente Wikipedia©).

Sacculina: Especie parásita de cirrípedo marino y que aparece en muchos crustáceos decápodos.

Virar: En ambientes marinos sinónimo de recoger cable.

GLOSARIO TÉCNICO

Actor: Toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo. Fuente ©Wikipedia.

AHAH: Acrónimo de **A**synchronous **H**TML **A**nd **H**ttp), **AHAH** es una técnica muy simple para crear elementos dinámicos en páginas Web utilizando JavaScript. Esto implica el uso de tecnologías [XMLHttpRequest](#) para extraer fragmentos de código [\(X\)HTML](#) que son insertados directamente en nuestra página Web, estando éstos formateados mediante hojas de estilo CSS.

Ajax: Acrónimo de **A**synchronous **J**avaScript **A**nd **X**ML (JavaScript asíncrono y [XML](#)), es una técnica de desarrollo web para crear aplicaciones interactivas o [RIA](#) (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Fuente ©Wikipedia.

Apache⁽²⁵⁾: es un servidor web HTTP de código abierto para plataformas [Unix](#) ([BSD](#), [GNU/Linux](#), etc.), [Microsoft Windows](#), [Macintosh](#) y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache⁽²⁵⁾ se desarrolla dentro del proyecto HTTP Server (httpd) de la [Apache Software Foundation](#). Apache⁽²⁵⁾ presenta entre sus aspectos destacables el ser altamente configurable, bases de datos de autenticación y negociado de contenido, pero carece de una interfaz gráfica que ayude en su configuración. Fuente © Wikipedia.

API: Referido a una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Fuente ©Wikipedia.

Cache: Referido en términos informáticos y en este contexto, a almacenamiento de los datos de uso habitual en el lado del cliente, a los que debe acceder una aplicación informática de manera frecuente. De este modo se evita el acceso continuo a los mismos.

Caso de uso: Secuencia de interacciones que se desarrollarán entre un sistema (una aplicación o un conjunto de aplicaciones) y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Fuente ©Wikipedia.

CSS: viene del [inglés](#) *Cascading Style Sheets* (hojas de estilo en cascada), del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en [HTML](#) o [XML](#) (y por extensión en [XHTML](#)). El [W3C](#) (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un documento de su *presentación*. Fuente ©Wikipedia.

Diagrama de actividad: Aquel que representa los flujos de trabajo paso a paso, de negocio y operacionales, de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general. Fuente ©Wikipedia.

Ficha CRC: Las **tarjetas CRC** (clase, responsabilidad y colaboración) son una metodología para el diseño de software orientado por objetos creada por Kent Beck y Ward Cunningham⁽³²⁾. Fuente ©Wikipedia.

Formulario Web: Permite al usuario introducir datos los cuales son enviados a un servidor para ser procesados. Los formularios web se parecen a los formularios de papel porque los internautas llenan dichos formularios usando casillas de selección, botones de opción, o campos de texto. Por ejemplo, los formularios web pueden ser usados para introducir datos de envío o datos de una tarjeta de crédito con el objetivo de solicitar un producto o bien ser utilizada para solicitar datos (p. ej., al buscar en un motor de búsqueda). Fuente ©Wikipedia.

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. Fuente ©Wikipedia.

HTTP: Hypertext Transfer Protocol o HTTP (en español *protocolo de transferencia de hipertexto*) es el protocolo usado en cada transacción de la World Wide Web.

IDE: Un entorno de desarrollo integrado (en inglés *Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Fuente ©Wikipedia.

JAVA⁽³¹⁾: es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Fuente ©Wikipedia.

JavaScript: es un lenguaje de programación interpretado, dialecto del estándar [ECMAScript](#). Se define como orientado a objetos, 3 basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Fuente ©Wikipedia.

Máquina virtual: es un software que emula a una computadora y puede ejecutar programas como si fuese una computadora real. Este software en un principio fue definido como "un duplicado eficiente y aislado de una máquina física". La acepción del término actualmente incluye a máquinas virtuales que no tienen ninguna equivalencia directa con ningún hardware real. Fuente ©Wikipedia.

Multidioma: En el contexto de este documento, se define cómo la capacidad de una Web de poder ser visualizada en diferentes idiomas, de una manera nativa. Siendo la aplicación web la que reconoce el idioma de navegación del usuario, mostrándole el contenido en su idioma, si este está disponible.

Open Source: O código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre. Fuente ©Wikipedia.

PHP⁽²⁷⁾: Acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*) que hace referencia a un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas [Qt](#) o [GTK+](#). Fuente ©Wikipedia.

Plugins: Aplicación complemento, que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. Su nombre proviene del inglés *plug-in* ("enchufable"), add-on (agregado), complemento, conector o extensión. Fuente ©Wikipedia.

Servidor: Un servidor web o servidor HTTP es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa. Fuente ©Wikipedia.

Sistema gestor de contenidos: del inglés *Content Management System*, (abreviado **CMS**) es un programa que permite crear una estructura de soporte ([framework](#)) para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

Site: O sitio web es una colección de páginas web relacionadas y comunes a un dominio de Internet o subdominio en la World Wide Web en Internet. Fuente ©Wikipedia.

URL: Localizador uniforme de recursos, más comúnmente denominado **URL** (sigla en inglés de *uniform resource locator*), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones, presentaciones digitales, etc. Fuente ©Wikipedia.

Usabilidad: El concepto de usabilidad (del inglés *usability*) es la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto. La usabilidad también puede referirse al estudio de los principios

que hay tras la eficacia percibida de un objeto. En nuestro caso concreto se refiere a la facilidad de uso de una herramienta informática. Fuente ©Wikipedia.

Web dinámica: Una **página web** es un documento o información electrónica adaptada para la *World Wide Web* que generalmente forma parte de un sitio web. Una página web está compuesta principalmente por información (sólo texto o módulos multimedia) El contenido de la página puede ser predeterminado («página web estática») o generado al momento de visualizarla o solicitarla a un servidor web («**página web dinámica**»). Las páginas dinámicas que se generan al momento de la visualización se hacen a través de lenguajes interpretados, generalmente JavaScript, y la aplicación encargada de visualizar el contenido es la que debe generarlo. Fuente ©Wikipedia.

REFERENCIAS

- (1) Senckenberg, World of Biodiversity: http://www.senckenberg.de/root/index.php?page_id=71&PHPSESSID=bjb7uh18isemii5pttjdn6nsugssnusa.
- (2) Instituto de Ciencias del Mar de Barcelona: <http://www.icm.csic.es/>.
- (3) Consejo Superior de Investigaciones Científicas: <http://www.csic.es/>.
- (4) Life in the World's Oceans: Diversity, Distribution and Abundance. Ed. A. McIntyre, 2010.
- (5) Discoveris of the Census of Marine Life: Making ocean life count. Paul Snelgrove, 2010.
- (6) Methods for the Study of deep-sea sediments, their functioning and biodiversity. R. Danovaro, 2010.
- (7) Deep-sea biodiversity: Pattern and scale. Rex & Etter, 2010.
- (8) Marine Ecology special issue: The roles of habitat heterogeneity in generating and maintaining biodiversity on continental margins. Levin, Sibuet, Gooday, Smith & Vanreusel (Eds), 2010.
- (9) Marine Ecology special issue: Recent advances in seamount ecology. Schlacher, Rowden, Dower & Consalvey, 2010.
- (10) Deep-Sea Research special issue: Deep-sea ecosystems of the equatorial African margin: the multidisciplinary BIOZAIRE program. Sibuet & Vangriesheim, 2009.
- (11) *PLoS ONE 2010: Marine Biodiversity and Biogeography - Regional Comparisons of Global Issues*:
<http://www.ploscollections.org/article/browseIssue.action;jsessionid=875F5EE8B935BFF82D7B5A2E4F383E22.ambra02?issue=info%3Adoi%2F10.1371%2Fissue.pcol.v02.i09>
- (12) [Microsoft Excel© 2010 Microsoft Corporation. Reservados todos los derechos.](#)
- (13) [Microsoft Access© 2010 Microsoft Corporation. Reservados todos los derechos.](#)
- (14) Publishing Network for Geoscientific & Environmental Data: <http://www.pangaea.de/>.
- (15) Sistema de Información Geográfica del Océano (OBIS): <http://www.iobis.org/es/node/212>.
- (16) Census of marine life: <http://www.coml.org/>
- (17) [Alfred Wegener Institute for Polar and Marine Research](#) (AWI), 27515 Bremerhaven, Germany.
- (18) [Center for Marine Environmental Sciences](#) (MARUM), University of Bremen, 28359 Bremen, Germany.
- (19) [MySQL © 2010, Oracle Corporation and/or its affiliates.](#)
- (20) Drupal © 2009 • [Drupal is a registered trademark of Dries Buytaert.](#)
- (21) Ubuntu © 2010 Canonical Ltd. [Ubuntu and Canonical are registered trademarks of Canonical Ltd.](#)
- (22) [Copyright © 2010 VMware, Inc. All rights reserved.](#)
- (23) [Microsoft Windows XP© Microsoft Corporation. Reservados todos los derechos.](#)
- (24) [MagicDraw © No Magic, Inc.](#) 2010 – 2011.
- (25) Apache, ©The Apache Software Foundation: <http://www.apache.org/>
- (26) NetBeans, © 2011, Oracle Corporation and/or its affiliates: <http://netbeans.org/>
- (27) PHP [Copyright © 2001-2011 The PHP Group.](#)
- (28) Light intensity determines temporal niche switching of behavioral activity in deep-water *Nephrops norvegicus* (Crustacea: Decapoda). Chiesa JJ, Aguzzi J, García JA, Sardà F, de la Iglesia HO. *J Biol Rhythms*. 2010 Aug;25(4):277-87.

- (29) Wikipedia © 2011: <http://es.wikipedia.org/>
- (30) *Coastal Ocean Observatory*. Instituto de Ciencias del Mar de Barcelona, CSIC: <http://coo.icm.csic.es/>
- (31) [Java Copyright© 1995, 2010, Oracle and/or its affiliates. All rights reserved.](#)
- (32) [A Laboratory For Teaching Object-Oriented Thinking](#). Kent Beck, Ward Cunningham, OOPSLA'89 Conference Proceedings, 1989, New Orleans, Louisiana. Volume 24, Number 10, October 1989
- (33) Universitat Oberta de Catalunya (UOC): <http://www.uoc.edu/>.

BIBLIOGRAFÍA

- (1) Drupal. Angela Byron, Adisson Beny, Nathan Haug, Jeff Eaton, James Walker y Jeff Robbins. Ediciones ANAYA-Multimedia – O'Reilly. ©Grupo ANAYA, 2010. ISBN: 978-84-415-2612-9
- (2) Drupal: Creating Blogs, Forums, Portals and Community Websites. David Mercer. ©Packt Publishing, 2006. ISBN: 1-904-811-80-9
- (3) PHP & MySQL Web Development: All-in-one desk reference for dummies®. Janet Valade, Tricia Ballad y Bill Ballad. Wiley Publishing, INC. 2008. ISBN: 978-0-470-16777-9