

JUNIO DEL 2014



XTREME
SnowBoard

SALESIANOS
ATOCHA

PROYECTO FIN DE CURSO

ALBERTO FUENTES SANCHEZ

ÍNDICE

1. INTRODUCCIÓN	2
OBJETIVO	2
JUSTIFICACIÓN.....	2
ANÁLISIS DE LO EXISTENTE.....	2
PROPUESTA DETALLADA	2
CRONOGRAMA.....	3
2. REQUISITOS Y MANUALES DE USUARIO.....	4
MODELO ENTIDAD/RELACIÓN	4
BASES DE DATOS (BBDD).....	5
INTERFAZ DE ADMINISTRACIÓN.....	9
SERVIDOR WEB.....	9
MONTAJE DE UN SERVIDOR WEB SOBRE UNA PLACA RASPBERRY PI (I)	10
INSTALACIÓN DE UN SERVIDOR WEB SOBRE UNA PLACA RASPBERRY PI (II)	18
ENRUTAMIENTO DE UN SERVIDOR WEB SOBRE UNA PLACA RASPBERRY PI (III)	22
3. DISEÑO	27
DISEÑO ARQUITECTONICO.....	27
PATRONES DE DISEÑO.....	28
FORMATO DE DISEÑO ESPECÍFICO.....	28
FORMATO DE DISEÑO DE DIFERENTES APARTADOS PÚBLICOS	33
FORMATO DE DISEÑO DEL SERVICIO: BUSCADOR DE HOTELES	35
4. CODIFICACIÓN	39
ENTORNO DE PROGRAMACION	39
LENGUAJES Y HERRAMIENTAS	39
ASPECTOS RELEVANTES DE LA IMPLEMENTACIÓN	40
CODIFICACIÓN DE REGISTRO	40
CODIFICACIÓN DE INICIO DE SESIÓN	47
CODIFICACIÓN DE SESIÓN	49
CODIFICACIÓN DEL BUSCADOR DE HOTELES	50
CODIFICACIÓN DE PAGO	53
CODIFICACIÓN DE ACCESO NO AUTORIZADO	55
5. PRUEBAS.....	57
6. CONCLUSIONES	57
7. POSIBLES AMPLIACIONES Y MEJORAS.....	58

1. INTRODUCCIÓN

OBJETIVO

Xtreme Snowboard es un portal web dedicado a las disciplinas deportivas del snowboard y del esquí. Dedicaremos nuestra labor a organizar herramientas de búsqueda online para la reserva de hoteles en determinadas estaciones de nieve, escuela de aprendizaje online, con video-tutoriales, e información y asesoramiento del material deportivo necesario para practicar estos deportes.

JUSTIFICACIÓN

Se pretende que todo ello sea accesible desde un portal web donde se focalizan todos los conocimientos necesarios para poder aprender, comprar y reservar nuestra estancia. La idea clave es no tener que salir de nuestro portal web para poder realizar todo tipo de gestiones relacionadas con este deporte. Para ello se tendrá acceso directo a una gran variedad de conocidas marcas y páginas web dedicadas al sector profesional del snowboard.

ANÁLISIS DE LO EXISTENTE

Hasta el momento no existe nada comparable en la red de internet. Nuestra idea abarca un mercado deportivo poco explotado dentro de un deporte en concreto que está empezando a practicarse cada vez más. La idea de “Buscador de viajes” es muy popular en internet, pero no existe nada en la red como “Buscador de viajes de snowboard y de esquí.” Cada invierno millones de personas tienen que organizar sus viajes deportivos y tienen que conformarse con buscadores que se dedican a todo tipo de viajes y sectores. Nuestra idea, no es solo que puedas encontrar un determinado plan de vacaciones, sino que también sea el mejor calidad/precio. El equipo de *Xtreme Snowboard* ha trabajado en una búsqueda selecta, para ofrecer lo mejor del mercado a precios muy atractivos para el cliente.

PROPUESTA DETALLADA

Reserva de Hoteles online:

La mayoría de proveedores de información serán todas y cada una de las estaciones con las que hayamos contratado nuestros servicios. Las estaciones de esquí nos proporcionan toda la información necesaria que necesitamos para desarrollar nuestro trabajo, y de la misma forma con los alojamientos de cada zona. Se pretende que una vez configurada la reserva realiza por el cliente se proceda a mostrar un listado de hoteles para que el cliente pueda elegir entre una gran variedad de alojamientos a

distintos precios. Ya elegida la estancia se procederá al registro de datos y al pago del importe estipulado en la web, no sin antes consultar todo el proceso de reserva antes de confirmar el pago.

Asesoramiento de material deportivo:

Incluiremos en nuestro portal web dos páginas dedicadas a lo último en material deportivo para la práctica de este deporte.

Escuela de aprendizaje:

Dispondremos de una conexión con Whistler BC (Canadá) con la empresa *SnowboardAddiction.com*, y otra conexión con la empresa *Thank You Shop*. Estas webs se dedican al video-aprendizaje de este deporte, ofreciendo una gran variedad de tutoriales a los que hacemos conexión directa.

Tienda:

La venta online del portal web estará asociada con dos reconocidas empresas a nivel nacional, siendo éstas: *Thank you Shop* y *Back In Black*, empresas dedicadas exclusivamente al deporte del snowboard y a la venta del correspondiente material deportivo. El cliente accederá a través de nuestra conexión y realizará sus compras externamente.

Contacto:

Se facilitará una página de contacto para los usuarios, estén o no registrados en nuestras bases de datos, para que puedan hacernos llegar sus dudas, inquietudes o preguntas.

Varios:

Ofertaremos un apartado de galería de video para hacer llegar a nuestros clientes los últimos *spots* del mercado relacionados con el snowboard. También se hablará de instalaciones para poder entrenar dicho deporte durante todo el año y se presentará en otro apartado a los miembros creadores del equipo *Xtreme Snowboard*.

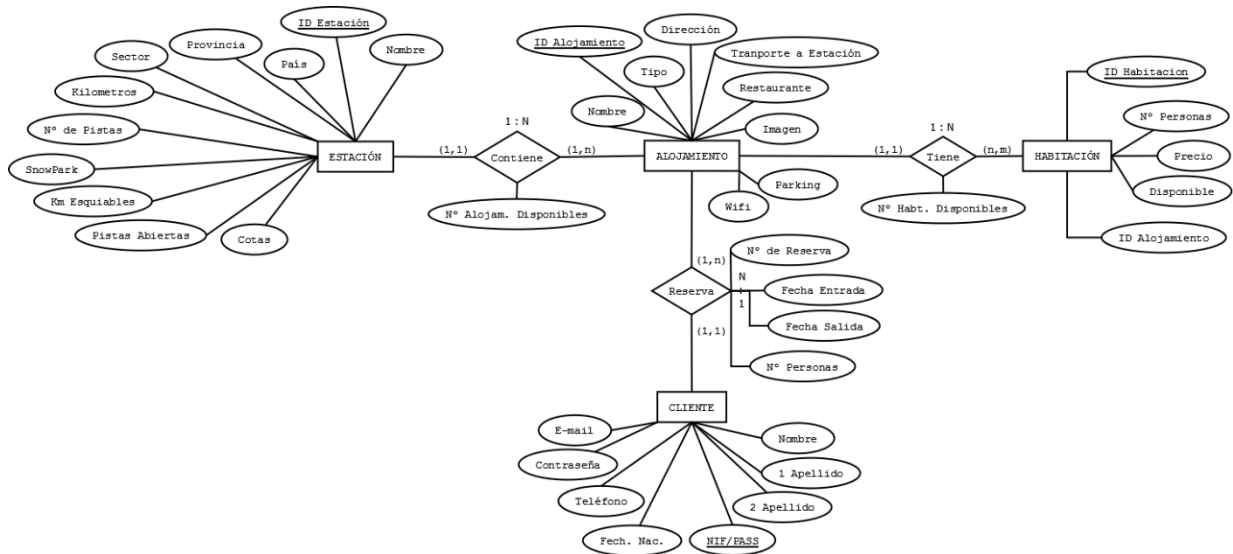
CRONOGRAMA

1. Creación del modelo Entidad/Relación para nuestra BBDD.
2. Creación de la BBDD.
3. Montaje, Instalación, y enrutamiento de un servidor web Linux
3. Estudio funcional y estructural de la web.
5. Estudio funcional del diseño a través de las Hojas de estilo.
6. Estudio de la Seguridad y la comprobación de posibles errores de conexión.
7. Desarrollo e implantación de diferentes lenguajes: HTML, CSS, PHP y JavaScript
8. Pruebas y Conclusiones.

2. REQUISITOS Y MANUALES DE USUARIO.

MODELO ENTIDAD/RELACIÓN

Necesitaremos plasmar el diseño de nuestro servicio de búsqueda de hoteles para posteriormente construir las bases de datos necesarias. Se ha diseñado con la herramienta de programa *Dia*. A continuación se define el siguiente modelo E/R:



Con propagación de clave:

ESTACIÓN (Nombre, País, Provincia, Sector, Km, Km_Esquiabiles, Pistas, Pistas_Abiertas, SnowPark, Cotas, ID_EST¹)

ALOJAMIENTO (Nombre, ID_ALOJ², Tipo, Dirección, Transporte, Restaurante, Parking, Wi-fi, Imagen)

HABITACIÓN (Num_Personas, Precio, Disponible, id_alojamiento, ID_HABITACION³)

CLIENTE (Nombre, 1apellido, 2apellido, NIF/PASS⁴, Fech_Nac, Telefono, E-mail, Contraseña)

CONTIENE (ID_EST¹, ID_ALOJ², Num_Aloj_Disponibles)

TIENE (ID_ALOJ², ID_HABITACION³, Num_Hab_Disponibles)

RESERVA (ID_ALOJ², NIF/PASS⁴, Num_Reserva, Fecha_Entrada, Fecha_Salida, Num_Personas)

BASES DE DATOS (BBDD)

Una vez tenemos nuestro modelo Entidad/Relación procedemos a crear las bases de datos de nuestro servicio en las que se almacenarán todos los datos necesarios para que el proceso sea lo más completo posible. Para ello se hará uso del lenguaje MySQL y de la consola de comandos de Windows y Linux.

1. Creación de las correspondientes tablas para la base de datos: *xtremesnowboard*

```
mysql> create database xtremesnowboard;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> use xtremesnowboard;  
Database changed
```

```
mysql> create table ESTACION(  
  -> Nombre varchar(20) not null,  
  -> Pais varchar(15) not null,  
  -> Provincia varchar(25) not null,  
  -> Sector varchar(20) not null,  
  -> Km int(4) not null,  
  -> Km_Esquiabiles int(4),  
  -> Pistas int(4) not null,  
  -> Pistas_Abiertas int(4),  
  -> SnowPark int(3),  
  -> Cotas varchar(10),  
  -> ID_EST int(10) not null primary key auto_increment  
  -> );  
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> create table ALOJAMIENTO(  
  -> Nombre varchar(25) not null,  
  -> ID_ALOJ int(10) not null primary key auto_increment,  
  -> Tipo varchar(20) not null,  
  -> Direccion varchar(80) not null,  
  -> Transporte varchar(2) check(Transporte in("SI","NO")),  
  -> Restaurante varchar(2) check(Restaurante in("SI","NO")),  
  -> Parking varchar(2) check(Parking in("SI","NO")),  
  -> Wifi varchar(2) check(Wifi in("SI","NO")),  
  -> Imagen varchar(255)  
  -> );  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table HABITACION(  
  -> ID_HAB int(10) not null primary key,  
  -> Num_Personas int(1) not null,  
  -> Precio float not null,  
  -> Disponible varchar(2) check(Disponible in("SI","NO")),  
  -> ID_Aloj int(10) not null  
  -> );  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table CLIENTE(  
  -> Nombre varchar(15) not null,  
  -> 1Apellido varchar(15) not null,  
  -> 2Apellido varchar(15) not null,  
  -> NIF_PASS varchar(15) not null primary key,  
  -> Fech_Nac date not null,  
  -> Telefono int(11) not null,  
  -> Email varchar(50) unique,  
  -> Password varchar(200)  
  -> );
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> create table EST_CONTIENE_ALOJ(  
  -> ID_EST int(10) references ESTACION,  
  -> ID_ALOJ int(10) references ALOJAMIENTO,  
  -> Num_Aloj_Disponibles int(4) not null,  
  -> primary key(ID_EST, ID_ALOJ, Num_Aloj_Disponibles)  
  -> );
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> create table ALOJ_TIENE_HAB(  
  -> ID_ALOJ int(10) references ALOJAMIENTO,  
  -> ID_HAB int(10) references HABITACION,  
  -> Num_Hab_Disponibles int(4) not null,  
  -> primary key(ID_ALOJ, ID_HAB, Num_Hab_Disponibles)  
  -> );
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> create table CLIENT_RESERVA_ALOJ(  
  -> ID_ALOJ int(10) references ALOJAMIENTO,  
  -> NIF_PASS varchar(15) references CLIENTE,  
  -> Num_Reserva int(10) auto_increment,  
  -> Fecha_Entrada date not null,  
  -> Fecha_Salida date not null,  
  -> Num_Personas int(1) not null,  
  -> primary key(Num_Reserva)  
  -> );
```

Query OK, 0 rows affected (0.03 sec)

2. Estructura visual de las tablas creadas en la base de datos: *xtremesnowboard*

```
mysql> show tables;
```

```
+-----+
| Tables_in_xtremesnowboard |
+-----+
| aloj_tiene_hab            |
| alojamiento              |
| client_reserva_aloj      |
| cliente                  |
| est_contiene_aloj        |
| estacion                 |
| habitacion               |
+-----+
7 rows in set (0.02 sec)
```

```
mysql> desc estacion;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Nombre         | varchar(20)   | NO   |     | NULL    |                |
| Pais           | varchar(15)   | NO   |     | NULL    |                |
| Provincia      | varchar(25)   | NO   |     | NULL    |                |
| Sector         | varchar(20)   | NO   |     | NULL    |                |
| Km             | int(4)        | NO   |     | NULL    |                |
| Km_Esquiables | int(4)        | YES  |     | NULL    |                |
| Pistas         | int(4)        | NO   |     | NULL    |                |
| Pistas_Abiertas | int(4)       | YES  |     | NULL    |                |
| SnowPark       | int(3)        | YES  |     | NULL    |                |
| Cotas          | varchar(10)   | YES  |     | NULL    |                |
| ID_EST         | int(10)       | NO   | PRI | NULL    | auto_increment |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.01 sec)
```

```
mysql> desc est_contiene_aloj;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID_EST         | int(10)       | NO   | PRI | 0        |                |
| ID_ALOJ        | int(10)       | NO   | PRI | 0        |                |
| Num_Aloj_Disponibles | int(4)       | NO   | PRI | NULL     |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```



```
mysql> desc alojamiento;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| Nombre     | varchar(25) | NO   |     | NULL    |            |
| ID_ALOJ    | int(10)    | NO   | PRI | NULL    | auto_increment |
| Tipo       | varchar(20) | NO   |     | NULL    |            |
| Direccion  | varchar(80) | NO   |     | NULL    |            |
| Transporte | varchar(2) | YES  |     | NULL    |            |
| Restaurante | varchar(2) | YES  |     | NULL    |            |
| Parking     | varchar(2) | YES  |     | NULL    |            |
| Wifi       | varchar(2) | YES  |     | NULL    |            |
| Imagen     | varchar(255) | YES  |     | NULL    |            |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> desc aloj_tiene_hab;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| ID_ALOJ        | int(10)   | NO   | PRI | 0        |            |
| ID_HAB         | int(10)   | NO   | PRI | 0        |            |
| Num_Hab_Disponibles | int(4)   | NO   | PRI | NULL     |            |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

```
mysql> desc habitacion;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| ID_HAB        | int(10)   | NO   | PRI | NULL     |            |
| Num_Personas  | int(1)    | NO   |     | NULL     |            |
| Precio        | float     | NO   |     | NULL     |            |
| Disponible    | varchar(2) | YES  |     | NULL     |            |
| ID_Aloj       | int(10)   | NO   |     | NULL     |            |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> desc client_reserva_aloj;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| ID_ALOJ        | int(10)   | YES  |     | NULL     |            |
| NIF_PASS       | varchar(15) | YES  |     | NULL     |            |
| Num_Reserva    | int(10)   | NO   | PRI | NULL     | auto_increment |
| Fecha_Entrada  | date      | YES  |     | NULL     |            |
| Fecha_Salida   | date      | YES  |     | NULL     |            |
| Num_Personas   | int(1)    | NO   |     | NULL     |            |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> desc cliente;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Nombre     | varchar(15)   | NO   |     | NULL    |       |
| 1Apellido  | varchar(15)   | NO   |     | NULL    |       |
| 2Apellido  | varchar(15)   | NO   |     | NULL    |       |
| NIF_PASS   | varchar(15)   | NO   | PRI | NULL    |       |
| Fech_Nac   | date          | NO   |     | NULL    |       |
| Telefono   | int(11)       | NO   |     | NULL    |       |
| Email      | varchar(50)   | YES  | UNI | NULL    |       |
| Password   | varchar(200)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

INTERFAZ DE ADMINISTRACIÓN

Necesitamos crear una interfaz web de administración de ámbito privado, para que los usuarios administradores gestionen la Base de Datos, de tal forma que puedan insertar, modificar o eliminar entradas de una forma amigable y directa sin tener que utilizar la línea de comandos del sistema. Para ello se establece una seguridad en la que sólo los usuarios con correo electrónico que contenga el dominio *@xtremesnowboard.com* podrán visualizar un botón en la esquina superior izquierda que, al accionarse, proporcionará el acceso a la interfaz web de administración. Con esto diferenciamos entre dos tipos de usuarios, el usuario estándar que sólo puede utilizar los servicios proporcionados por los usuarios administradores, y el usuario administrador que puede manejar todo el servicio que ofrece al usuario estándar. *(Véase también el apartado de **Diseño de la Interfaz de Administración** y el apartado de **Codificación de la Interfaz de Administración**).*

SERVIDOR WEB

Como último requisito necesitaremos proporcionar servicio a través de un servidor web, donde se almacenará a su vez la BBDD. Para ello hemos construido un servidor casero utilizando una placa base Raspberry Pi. A continuación se muestra todo el proceso de implantación llevado a cabo a través de los siguientes manuales de usuario creados para su explicación.

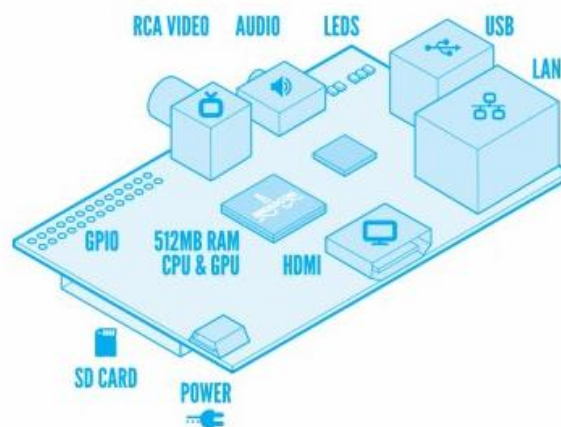
MONTAJE DE UN SERVIDOR WEB SOBRE UNA PLACA RASPBERRY PI (I)

Se pueden implementar infinidad de funciones y aplicaciones sobre una placa Raspberry Pi. Vamos a Instalar y configurar un sistema operativo para nuestro futuro servidor web del proyecto de fin de grado. A continuación, para los que no saben qué es Raspberry, explicaremos todo detalladamente para entender la funcionalidad que queremos conseguir con Raspberry Pi.

1. Raspberry Pi.

Raspberry Pi es una placa computadora (SBC) de bajo costo desarrollada en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño de la Raspberry Pi incluye:



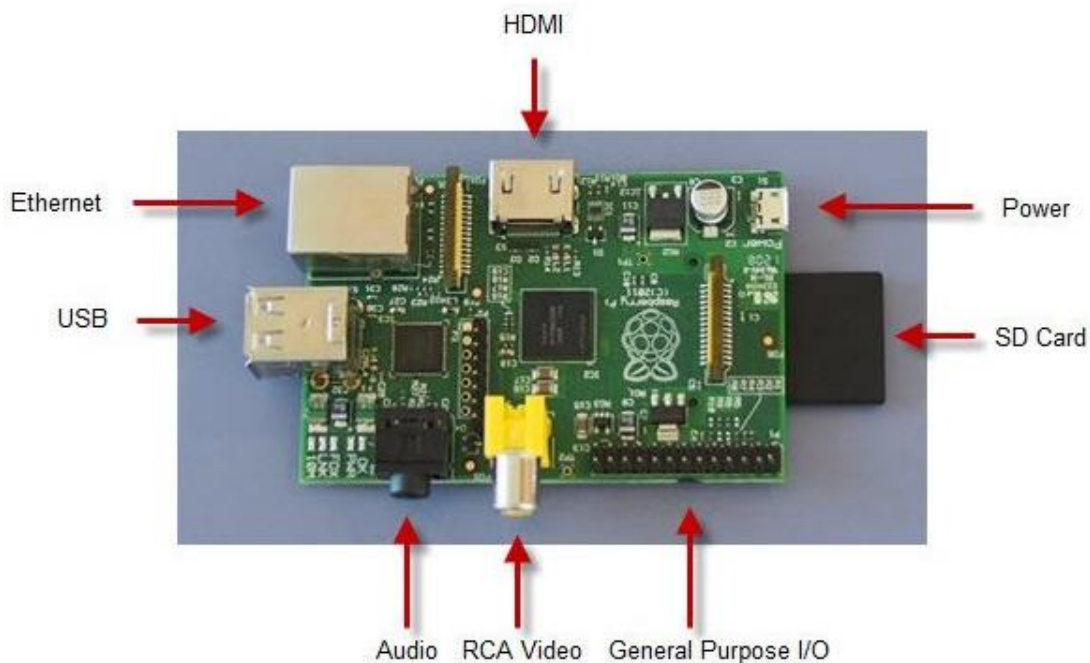
- System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) **ARM1176JZF-S** a 700 MHz.
- El firmware incluye unos modos Turbo para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía.
- Un procesador gráfico (GPU) VideoCore IV.
- 512 MB de memoria RAM.
- Salida RCA de Video (minijack).
- Salida de Audio (minijack).
- 2 puertos USB 2.0
- Puerto RJ-45 Ethernet.
- Puerto HDMI (Salida de Audio y Video simultáneo).
- Ranura para tarjetas SD.

Todo ello equivale en la práctica a un ordenador con unas capacidades gráficas similares a la XBOX de Microsoft y con la posibilidad de reproducir vídeo en 1080p.

La Raspberry Pi posee una conexión ethernet 10/100 y, si bien es cierto que podría echarse en falta una conexión Wi-Fi, gracias a los dos puertos USB incluidos podremos suplir dicha carencia con un adaptador Wi-Fi USB de terceros si lo necesitamos. Los puertos tienen una limitación de corriente, por lo que si queremos conectar discos duros u otro dispositivos tendremos que pensar en hacerlo a través de un hub USB con alimentación.

Para alimentar nuestra Raspberry Pi necesitaremos un alimentador entre 0.7 A y 1.5 A. El adaptador no viene incluido en la compra de la placa por lo que deberemos adquirir uno. Dicho alimentador tiene que tener un conector micro-USB para poder conectarlo a la Raspberry Pi.

Nuestra tarjeta SD será donde se almacenará nuestro sistema operativo y todos nuestros datos. Es recomendable adquirir una tarjeta SD de más de 4 Gb de memoria.



2. Sistemas Operativos en Raspberry Pi.

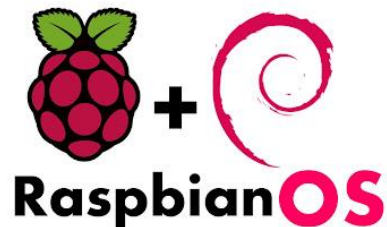
La Raspberry Pi se entrega sin sistema operativo, éste deberemos instalarlo sobre una tarjeta SD que introduciremos en la ranura SD. La mayoría de sistemas operativos soportados son distribuciones Linux/Debian.

Esta es una lista de sistemas operativos que funcionan, se han soportado, o están en proceso de ser soportados a Raspberry Pi:

- **Sistemas operativos completos:**
 - AROS
 - Linux
 - Android
 - Arch Linux ARM
 - Debian Whezzy Soft-Float, versión de Debian sin soporte para coma flotante por hardware
 - Firefox OS
 - Gentoo Linux
 - Google Chromium OS
 - Open webOS
 - PiBang Linux , distribución Linux derivada de Raspbian con diferente escritorio y aplicaciones
 - Pidora, versión Fedora Remix optimizada
 - QtonPi, distribución linux con un framework de aplicaciones multiplataforma basado en Qt framework
 - Raspbian , versión de Debian Whezzy para ARMv6 con soporte para coma flotante por hardware
 - Slackware ARM, también conocida como ARMedslack
 - Plan 9 from Bell Labs
 - RISC OS 5
 - Unix
 - FreeBSD
 - NetBSD
- **Distribuciones ligeras multipropósito:**
 - Moebius, distribución ligera ARM HF basada en Debian que usa el repositorio de Raspbian y que cabe en una tarjeta SD de 1GB, usa pocos servicios y está optimizada para usar poca memoria.
 - Squeezed Arm Puppy, una versión de Puppy Linux (Puppi) para ARMv6 (sap6) específicamente para Raspberry Pi.
- **Distribuciones ligeras de único propósito:**
 - Instant WebKiosk, sistema operativo con solo un navegador
 - IPFire
 - OpenELEC
 - Raspbmc
 - Xbian

3. Raspbian en Raspberry Pi.

A continuación vamos a detallar como hemos instalado el sistema operativo **Raspbian** en la tarjeta SD de nuestra Raspberry Pi. Hemos elegido una distribución basada en Debian por su fácil configuración para el montaje de un futuro servidor web para nuestro proyecto de fin de grado.



Hardware necesario

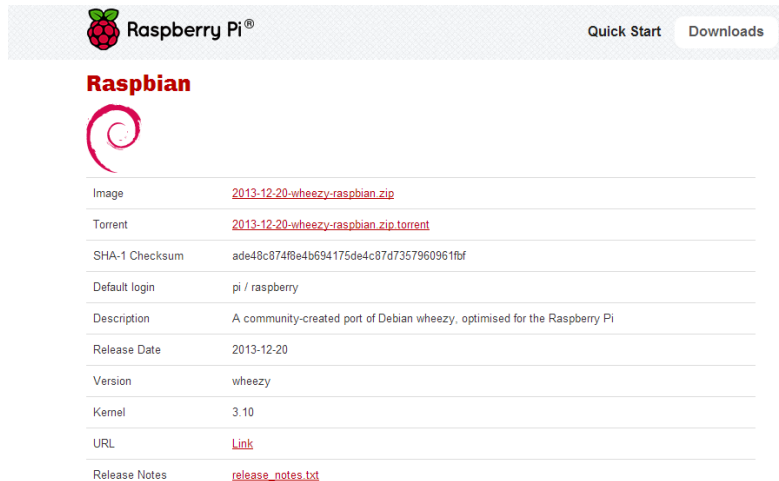
- Raspberry Pi.
- SD de 2Gb de capacidad como mínimo.
- Fuente de alimentación con conector micro-USB que suministre >750 mA.
- Cable HDMI o RCA para mandar la salida de vídeo y audio al televisor.
- Cable Ethernet (necesario si queremos conectarnos en red).
- Ratón y teclado.

Software necesario

- Win32 Disk imager, veremos cómo descargarlo e instalarlo.
- MD5 & SHA-1 Checksum Utility, veremos cómo descargarlo e instalarlo.
- Sistema Operativo Raspbian “Wheezy”, veremos cómo descargarlo e instalarlo.

COMENZAMOS...**DESCARGA DEL SISTEMA OPERATIVO RASPBIAN:**

1. Debemos ir a la siguiente página <http://www.raspberrypi.org>
2. Nos dirigimos a la sección **Downloads**, buscamos **Raspbian** y descargaremos el archivo **.zip**



Raspbian	
Image	2013-12-20-wheezy-raspbian.zip
Torrent	2013-12-20-wheezy-raspbian.zip.torrent
SHA-1 Checksum	ade48c874f0e4b694175de4c87d7357960961bf
Default login	pi / raspberry
Description	A community-created port of Debian wheezy, optimised for the Raspberry Pi
Release Date	2013-12-20
Version	wheezy
Kernel	3.10
URL	Link
Release Notes	release_notes.txt

Se nos abrirá una página nueva al mismo tiempo donde empezará la descarga. No cerrar la página ya que necesitaremos de su contenido para verificar la correcta descarga del archivo.

DESCARGA DE MD5 & SHA-1 CHECKSUM UTILITY:

1. Ir al siguiente enlace http://descargar.cnet.com/MD5-SHA-Checksum-Utility/3000-2092_4-10911445.html
2. Hacer clic en **“Descargar Ahora”**.



Download.com
Powered by CNET

Búsqueda de descargas

Windows Mac iOS Android

Start Download
3 steps for faster install & scan

Free Download

1. Click Free Download
2. Run the quick scan
3. Fix the errors. - Regclean Pro

Inicio > Software para Windows > Software de seguridad > Software de encriptado > MD5 & SHA Checksum Utility

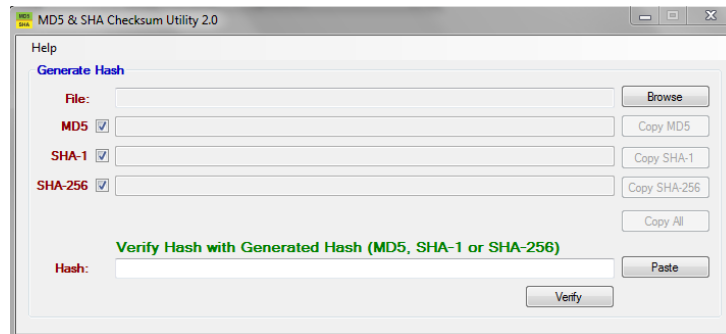
MD5 & SHA Checksum Utility

Descargar ahora
Descarga segura de CNET

Reseña de los editores de CNET
por el personal de CNET en diciembre de 01, 2009

Start Download
3 steps for faster install & scan

Free Download

VERIFICACIÓN DEL SISTEMA OPERATIVO:1. Ejecutar el programa descargado **MD5 & SHA-1 Checksum Utility**:

2. Hacemos clic en **Browse** y vamos a buscar el archivo .zip, descargado anteriormente, que contiene Raspbian.

3. Volvemos al navegador y recuperamos la página que se nos había abierto cuando habíamos empezado a descargar el archivo. Copiamos y pegamos el SHA-1 Checksum:

File: 2012-12-16-wheezy-raspbian.zip

File Size: 483.61 MiB

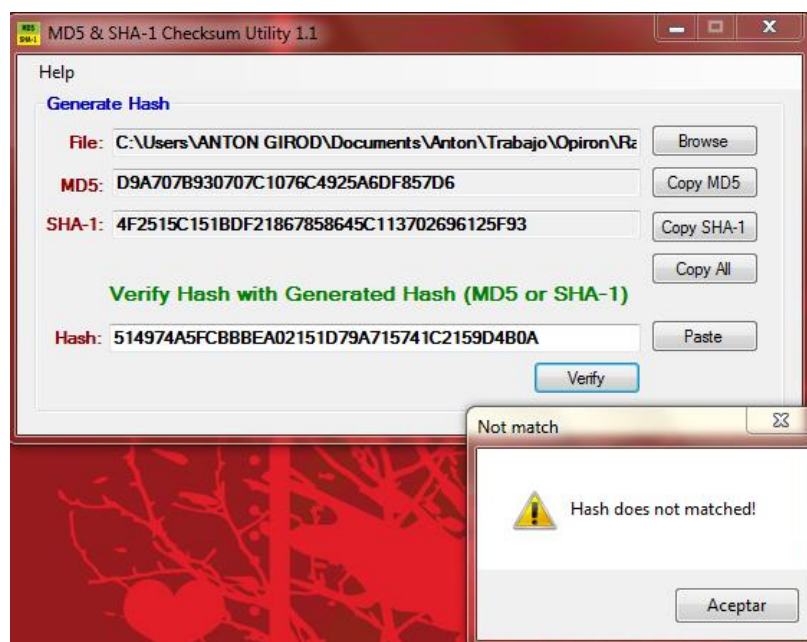
You will be redirected in 5 seconds. Don't want to wait? Use a [Direct Link](#).

SHA-1 Checksum: 514974a5fcbbbea02151d79a715741c2159d4b0a

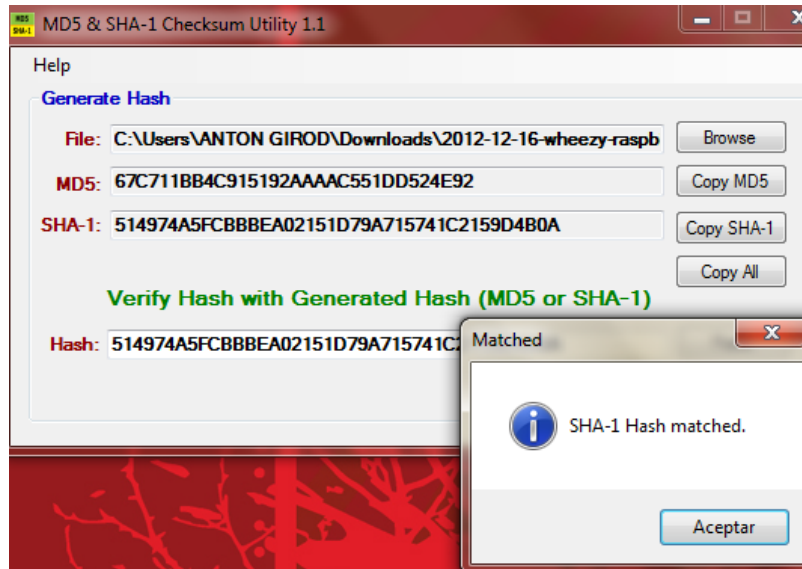
We recommend that you verify the image with the SHA-1 checksum provided above. Instructions for this are [Here](#).

Tenemos que copiar y pegar el SHA-1 Checksum que aparece.

4. Finalmente para comparar el hash le damos al botón **Verify**.



En el ejemplo anterior vemos dos Hash que no corresponden, si esto sucede tenemos que volver a descargarnos el sistema operativo y seguir los pasos de nuevo hasta este punto. En cambio, en la siguiente imagen vemos como sí corresponden y por lo tanto tenemos un S.O. correcto:

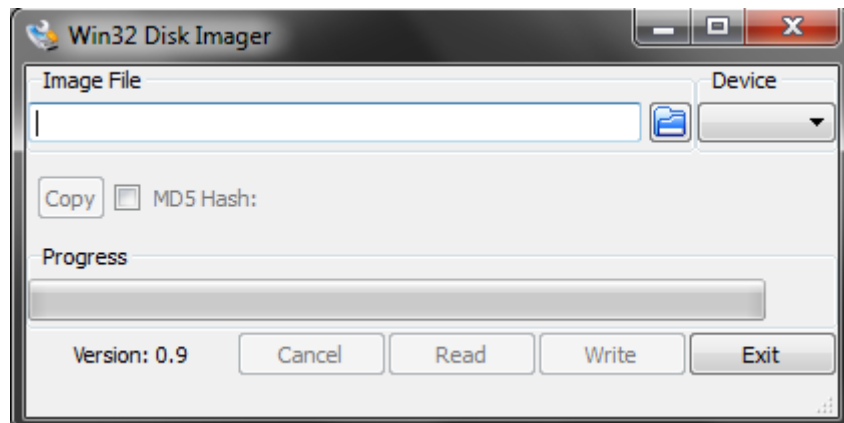


DESCARGA E INSTALACIÓN DE Win32DiskImager:

1. Descargamos la aplicación Win32DiskImager del siguiente enlace <http://www.softpedia.es/programa-Win32-Disk-Imager-173006.html>.
2. Para instalarlo, simplemente descomprimos el archivo descargado en la ruta que queramos.

INSTALACIÓN DEL SISTEMA OPERATIVO RASPBIAN:

1. Descomprime el archivo .zip que contiene el sistema operativo Raspbian.
2. Inserta la SD Card al PC.
3. Abrimos Win32 Disk Imager como administrador, desplegamos la pestaña **Device** y seleccionamos nuestra tarjeta SD recién insertada en nuestro equipo. Hacemos clic en el icono de la **carpeta de color azul** y seleccionamos el archivo Raspbian descomprimido en el paso 1.



4. Hacemos clic en **Write** y esperamos a que la aplicación escriba en la SD el Sistema Operativo Raspbian.
5. Si todo ha ido bien, una vez terminado el proceso tendremos el Sistema Operativo Raspbian instalado en la SD.
6. Finalmente, ya podemos insertar nuestra tarjeta SD en Raspberry Pi.

PRIMER ARRANQUE:

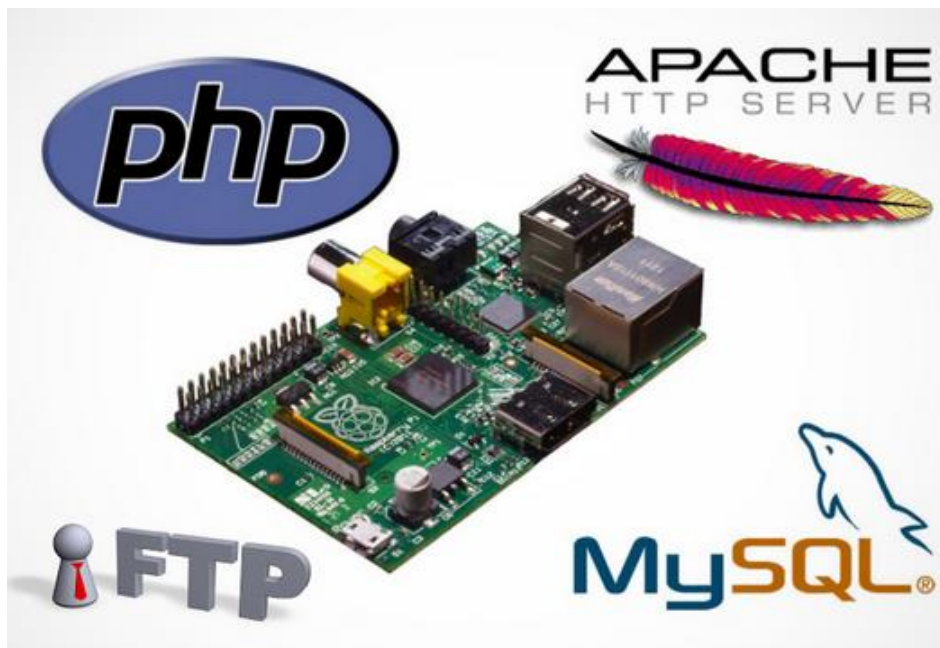
1. En el primer arranque aparecerá la ventana de configuración de **Raspi-config**, donde podremos cambiar parámetros como la hora local. Seleccionar la opción **expand_rootfs** y decir **'yes'** para resetear.
2. La raspberry pi se reiniciará y tendremos que logearnos. Insertaremos: **Type:** *'pi'* y **password:** *'ElQueQueramos'*.
3. En ese momento las líneas de comando aparecerán como **pi@raspberrypi ~ \$**.
4. Podemos arrancar la interfaz gráfica escribiendo en la línea de comandos: **'startx'**.

INSTALACIÓN DE UN SERVIDOR WEB SOBRE UNA PLACA RASPBERRY PI (II)

A continuación vamos a explicar cómo hemos instalado un **servidor web completo** en Raspberry Pi para poder acceder a nuestro proyecto web de fin de grado desde nuestra propia LAN. Nuestro servidor web contará con las siguientes aplicaciones de servicio:

- Apache Server
- PHP5
- MySQL
- FTP

Bajo nuestro punto de vista, un servidor web no está completo si carece de alguno de estos elementos.



1. Configurar la red y Actualizar Raspberry Pi.

1. En primer lugar comprobaremos que contamos con **la última versión de Raspbian Wheezy** que podemos descargar desde la página de descargas de Raspberry Pi (<http://www.raspberrypi.org/downloads>).

2. A continuación comenzaremos con el proceso de instalación. En primer lugar **actualizaremos todo lo que se pueda** actualizar mediante los siguientes comandos:

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

3. Ahora asignaremos una IP fija a nuestra Raspberry dentro de nuestra LAN para poder tener acceso a nuestro servidor de forma remota desde cualquier equipo de nuestra red y poder mandar peticiones a dicho servidor. Para ello editaremos el archivo de configuración de interfaces de red con el siguiente comando:

```
$ sudo nano /etc/network/interfaces
```

```
GNU nano 2.2.6          Fichero: /etc/network/interfaces
auto lo
iface lo inet loopback
#iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

auto eth0
iface eth0 inet static
address 192.168.1.50
netmask 255.255.255.0
gateway 192.168.1.1
nameserver 80.58.61.250 80.58.61.254

Nombre del fichero a escribir: /etc/network/interfaces
^G Ver ayuda      M-D Formato DOS  M-A Añadir      M-B Respalda fich
^C Cancelar      M-M Formato Mac  M-P Anteponer
```

2. Instalación de Apache y PHP en Raspberry Pi.

1. Una vez tengamos actualizado el sistema procederemos a instalar los complementos necesarios para nuestro servidor. Lo primero será **Apache Server** y **PHP5**, los cuales instalaremos con el siguiente comando:

```
$ sudo apt-get install apache2 php5 libapache2-mod-php5
```

2. Ahora podremos acceder mediante un navegador a la web por defecto de Apache a través de la **IP de nuestra Raspberry Pi** (192.168.1.50) o escribiendo en la barra de direcciones **“localhost”**. Si falla reiniciaremos el servicio con el siguiente comando:

```
$ sudo service apache2 restart
```

3. Instalación de MySQL en Raspberry Pi.

1. El siguiente complemento que instalaremos será **MySQL** para poder gestionar bases de datos. Lo haremos mediante el siguiente comando:

```
$ sudo apt-get install mysql-server mysql-client php5-mysql
```

4. Instalación de un servidor FTP en Raspberry Pi.

1. Para finalizar la instalación añadiremos un **servidor FTP** a nuestra Raspberry Pi. En primer lugar **cogeremos la propiedad del directorio raíz de la web** mediante el siguiente comando:

```
$ sudo chown -R pl /var/www
```

2. Posteriormente **instalaremos el servidor vsftpd** introduciendo el siguiente comando:

```
$ sudo apt-get install vsftpd
```

3. A continuación **configuraremos el servidor FTP para obtener mayor seguridad**, para ello editaremos el archivo **vsftpd.conf**.

```
$ sudo nano /etc/vsftpd.conf
```

4. Cambiaremos lo siguiente:

- De **anonymous_enable=YES** a **anonymous_enable=NO**
- Quitaremos la **#** de las líneas **local_enable=YES** y **write_enable=NO**
- Añadiremos la línea **force_dot_files=YES** al final del documento

```

GNU nano 2.2.6          Fichero: /etc/vsftpd.conf
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=NO
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
Nombre del fichero a escribir: /etc/vsftpd.conf
^G Ver ayuda      M-D Formato DOS  M-A Añadir      M-B Respalda fich
^C Cancelar      M-M Formato Mac  M-E Anteponer

```

```

GNU nano 2.2.6          Fichero: /etc/vsftpd.conf
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
#
# This option should be the name of a directory which is empty. Also, the
# directory should not be writable by the ftp user. This directory is used
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
secure_chroot_dir=/var/run/vsftpd/empty
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/private/vsftpd.pem
force_dot_files=YES
Nombre del fichero a escribir: /etc/vsftpd.conf
^G Ver ayuda      M-D Formato DOS  M-A Añadir      M-B Respalda fich
^C Cancelar      M-M Formato Mac  M-E Anteponer

```

5. Para terminar reiniciaremos el servicio.

\$ sudo service vsftpd restart

6. Ya tenemos nuestro servidor web completo instalado correctamente en Raspberry.

ENRUTAMIENTO DE UN SERVIDOR WEB SOBRE UNA PLACA RASPBERRY PI (III)

En esta parte veremos cómo damos acceso al servidor desde el exterior de nuestra red. Obtendremos un dominio personalizado con el que acceder a nuestra red e intentaremos aportar un poco de seguridad al sistema.

1. Configuración de DNS dinámico con No-IP en Raspberry Pi.

Hasta ahora hemos creado un servidor doméstico con nuestra Raspberry Pi, y puesto que lo más habitual en nuestros hogares es disponer de una conexión con IP dinámica deberemos conseguir una forma amigable de conectarnos desde el exterior, ya que la IP que te puedas apuntar hoy puede que no te sirva mañana.

Hay multitud de servicios orientados a solucionar este problema, uno de los más utilizados es el de No-IP que será el que utilizaremos a continuación.

Lo primero que debemos hacer es registrarnos en su website <http://www.noip.com/> y crear un nuevo hostname del tipo **"A"** con un nombre que podamos recordar y asignando el dominio que prefiramos de la lista.

Hecho esto tenemos nuestro nombre de dominio creado, pero necesitamos **asociarlo a nuestra red**, y además hacer que **nuestra Raspberry Pi actualice ese registro cada cierto intervalo de tiempo** para que, cuando nuestra dirección IP pública cambie, el hostname de No-IP se actualice.

1. Nos dirigimos al directorio desde el que instalaremos el programa:

```
$ cd /usr/local/src
```

2. Descargamos el programa:

```
$ sudo wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

3. Lo descomprimos e instalamos:

```
$ sudo tar -xvzf noip-duc-linux.tar.gz
```

```
$ cd no-ip-2.1.9-1
```

```
$ sudo make
```

```
$ sudo make install
```

Aquí se nos generará el archivo de configuración por defecto **/usr/local/etc/no-ip2.conf** con los datos que se nos preguntarán.

4. Los rellenamos con nuestros datos de No-IP y ya tendremos todo listo. Ahora con ejecutar el archivo `/usr/local/bin/noip2` se nos actualizará el registro hostname en no-ip cada "X" intervalo que hayamos configurado.

5. Lo último que deberemos hacer será crear un simple script que se **ejecute tras cada reinicio** para que llame a este ejecutable. En el directorio `/usr/local/src/noip-2.1.9-1` se nos proporciona información sobre cómo hacerlo en perfecto castellano en el archivo **LEEME.PRIMERO** debajo del apartado **COMO INICIAR EL CLIENTE**.

Así que procederemos a crear el archivo en init.d con la siguiente instrucción:

```
$ sudo nano /etc/init.d/noip2
```

Y escribimos el siguiente contenido en dicho archivo:

```
GNU nano 2.2.6          Archivo: /etc/init.d/noip2
#!/bin/sh
case "$1" in
start)
echo "Iniciando el servicio noip2..."
/usr/local/bin/noip2
;;
stop)
echo -n "Apagando el servicio noip2..."
killproc -TERM /usr/local/bin/noip2
;;
*)
echo "Uso: $0 {start|stop}"
exit 1
esac
exit 0

Nombre del fichero a escribir: /etc/init.d/noip2
^G Ver ayuda      M-D Formato DOS  M-A Añadir       M-B Respalda fich
^C Cancelar      M-M Formato Mac  M-P Anteponer
```

6. Guardamos y procedemos a darle permisos de ejecución al script así como actualizar la configuración de arranque con este script:

```
$ sudo chmod 755 /etc/init.d/noip2
```

```
$ sudo update-rc.d noip2 defaults
```

7. Podemos reiniciar la Raspberry Pi para comprobar que efectivamente se ejecuta el script.

```
$ sudo shutdown -r now
```

2. Añadiendo seguridad a Raspberry Pi.

Tenemos que ser muy conscientes de lo que estamos haciendo, estamos abriéndonos la puerta de entrada al dispositivo desde el exterior de nuestra red local, por lo que los riesgos de seguridad se disparan. Hay algunas medidas de seguridad que podemos aplicar para conseguir un plus de seguridad en el dispositivo. Estas medidas no son infranqueables, nada en el mundo de la informática lo es, pero podemos intentar protegernos de los ataques más frecuentes.

SEGURIDAD EN LOS USUARIOS

Este es el primero de todos los pasos. Hay que ser consciente que si no disponemos de una contraseña robusta para nuestros usuarios estamos dejando la puerta abierta a nuestro sistema, cosa que no queremos. Debemos establecer una contraseña robusta a nuestro dispositivo conteniendo letras, números, caracteres especiales, etc.

Una vez hemos determinado la contraseña que queremos usar podemos actualizarla con un simple:

```
$ passwd
```

SEGURIDAD EN SSH

El primer servicio al que probablemente queremos acceder una vez abierta la comunicación con el exterior es a ssh, para así poder acceder a nuestro dispositivo desde donde queramos. Como este servicio provee acceso completo a nuestro sistema deberemos realizar algunas tareas para asegurar el servicio. Podemos acceder a su archivo de configuración desde:

```
$ sudo nano /etc/ssh/sshd_config
```

Hay muchas cosas que nos puede interesar cambiar en este archivo, pero básicamente tocaremos dos. Cambiaremos el puerto por defecto ya que como anteriormente he dicho hay que huir de las configuraciones por defecto. Así que nos dirigimos al campo "Port" y cambiamos el 22 por el que nos plazca:

```
GNU nano 2.2.6      Fichero: /etc/ssh/sshd config      Modificado
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 2014
```

Por otro lado prohibiremos al usuario root la conexión ya que en el supuesto caso de que se consiguiera acceso con este usuario por un atacante estaríamos totalmente vendidos. Así que nos dirigimos a PermitRootLogin y cambiamos el “yes” por “no”.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
```

A partir de ahora para conectaros vía ssh a la Raspberry Pi habrá que especificar el puerto, ya sea rellenando el campo correspondiente en PuTTY o con la opción -p en GNU/Linux:

```
$ ssh pi@raspberrypi -p 2014
```

EVITAR ATAQUES DE FUERZA BRUTA CON FAIL2BAN

Fail2ban es una aplicación que monitoriza los logs de los principales servicios a instalar en un servidor en busca de intentos fallidos de login. Al detectarse una cantidad “X” de intentos (configurable) desde una misma IP se procede a bloquear el acceso durante “Y” tiempo (configurable).

Para instalarlo:

```
$ sudo apt-get install fail2ban
```

Fail2ban viene con una serie de servicios ya configurados, facilitando extremadamente su puesta en marcha.

Su archivo de configuración se encuentra en **/etc/fail2ban/jail.conf** y procedemos a editar:

```
$ sudo nano /etc/fail2ban/jail.conf
```

En la sección **[DEFAULT]** encontramos las opciones:

```
ignoreip = 127.0.0.1/8 # IP's a ignorar. Podemos añadir el rango local
bantime = 172800 # Tiempo en segundos del bloqueo.
maxretry = 5 # Cantidad de intentos fallidos permitidos.
```

Ahora nos dirigimos a la sección **JAILS** donde encontraremos **[ssh]**. Nos aparecerá una configuración por defecto preparada para funcionar con ssh y que modificaremos para cumplir con nuestros intereses:

```
enabled = true # Indicaremos que empiece a filtrar ssh.
port = 6784 # Aquí pondremos el puerto configurado anteriormente.
port = 2014 en nuestro caso.
```

Hecho esto reiniciamos el servicio con la siguiente instrucción:

\$ sudo service fail2ban restart

APERTURA DE PUERTOS

Hecho esto debemos **redirigir el puerto 2014** en nuestro router apuntando a nuestro servidor Raspberry Pi. Es decir debemos abrir el puerto especificado en nuestro router.

Raspberry	2014	2014	TCP	2014	2014	192.168.1.50	ppp0.6	<input type="checkbox"/>
-----------	------	------	-----	------	------	--------------	--------	--------------------------

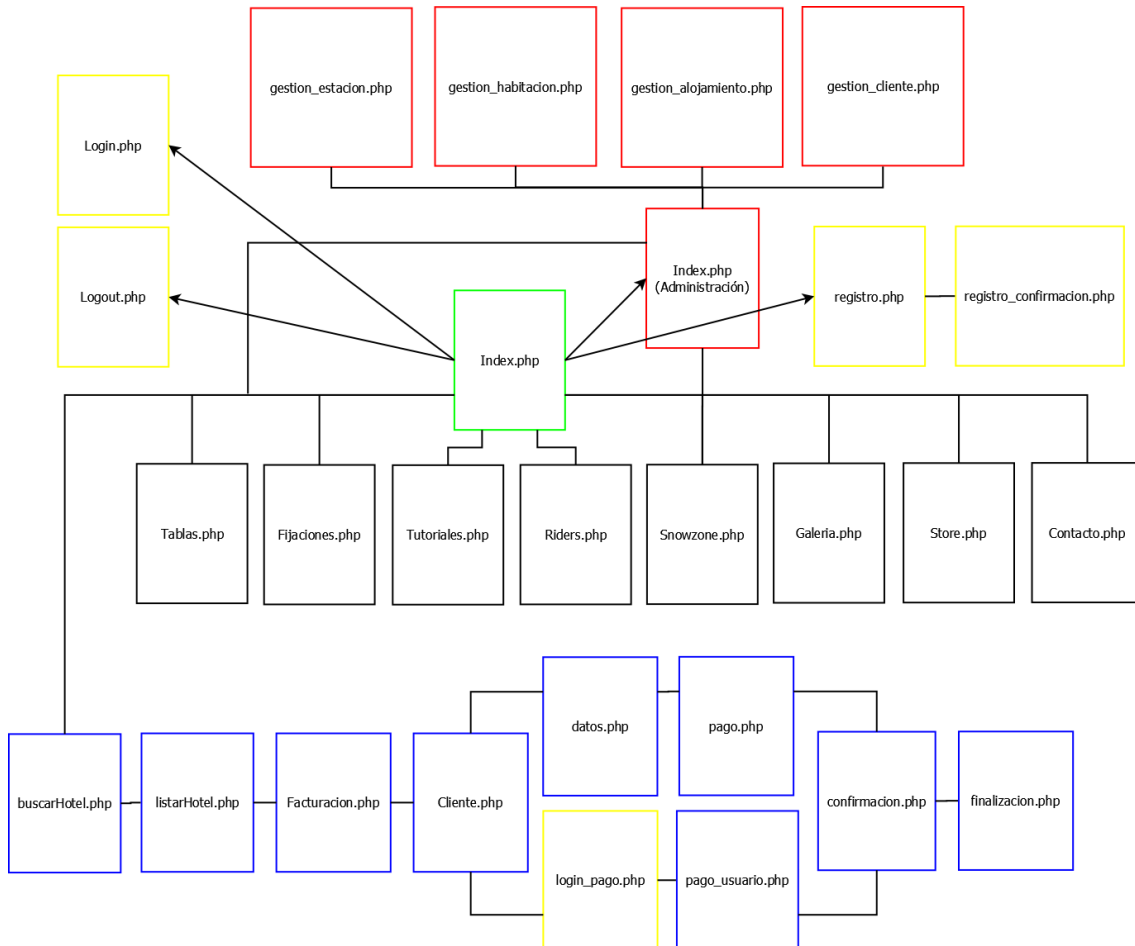
Hay que tener en cuenta que estas medidas **no nos protegerán ante todo lo que acecha al otro lado de nuestro router**, pero sí que evitaremos caer ante ataques automatizados de fuerza bruta, que al final son los más fáciles de llevar a cabo.

Es interesante ir **revisando periódicamente los logs del sistema**, situados en **/var/log/** para detectar posibles intrusiones a nuestro sistema.

3. DISEÑO

DISEÑO ARQUITECTONICO

A continuación se muestra el diseño arquitectónico del total de páginas contenidas en la interfaz web del portal *Xtreme Snowboard*.



Negro: varios apartados de información, asesoramiento, video, imagen y tienda que contiene el *web site*. Esta sección es de acceso público, usuario invitado.

Verde: **página principal** de la cual derivan los diferentes apartados; los servicios; el acceso a la interfaz web de administración; y los procesos de registro y *logueo* de los usuarios en el sistema. Esta sección es de acceso público.

Amarillo: **procesos** de registro; inicio de sesión y cierre de sesión de los usuarios. Esta sección es donde se implanta la máxima seguridad.

Azul: **servicios** que proporciona la web, en este caso el **buscador** de viajes online. Esta sección es común a los usuarios estándar y a los usuarios administradores

Rojo: interfaz de **administración** de ámbito privado exclusiva para los usuarios administradores.

PATRONES DE DISEÑO

A continuación vamos a explicar el diseño de la interfaz visual de cada área de la web.

FORMATO DE DISEÑO ESPECÍFICO

INDEX.PHP*

Usuario Invitado

Esta es la página principal de acceso público. Si el usuario se identifica en el cuadro de acceso situado en la parte inferior izquierda, podrá acceder a la parte de usuario estándar.



Usuario Estándar

El cuadro de acceso desaparece evitando que el usuario pueda *reloguearse*, y le da la opción al usuario de cerrar su sesión en el sistema cuando desee.



Usuario Administrador

Como vemos cuadro de acceso situado en la parte inferior izquierda ha desaparecido evitando que el usuario pueda *reloguearse* ya sea estándar o administrador.

En el caso de que se identifique como usuario administrador, podrá acceder a la página principal de control. Para ello se establece una seguridad en la que sólo los usuarios con correo electrónico que contenga el dominio *@xtremesnowboard.com* podrán visualizar un botón en la esquina superior izquierda que, al accionarse, proporcionará el acceso a la interfaz web de administración como mostramos en el apartado siguiente (*GESTION_(apartado).PHP**).



(Esta parte tiene un aspecto relevante de codificación de seguridad) Véase **Codificación de Sesión**.

GESTION (apartado).PHP*

Este es el formato establecido en las páginas de gestión de **estación, alojamiento, habitación y cliente**. Esta sección es exclusiva para los usuarios administradores, se da acceso a gestionar la Base de Datos, de tal forma que puedan insertar, modificar o eliminar entradas de una forma amigable y directa sin tener que utilizar la línea de comandos del sistema. Si cerramos sesión se direccionará inmediatamente a la página principal, manteniendo la integridad de las bases de datos del sistema.

The screenshot shows a web application interface for 'XTREME snowBoard'. The main content area is titled 'GESTIÓN DE ESTACIÓN' and is divided into two sections: 'ACTUALIZACIÓN DE LA ESTACION' and 'INSERTAR NUEVA ENTRADA'.

ACTUALIZACIÓN DE LA ESTACION

Form fields:

- Nombre de la Estacion:*
- Km Esquiabiles:
- Pistas Abiertas:
- Cotas:

Buttons: ACTUALIZAR, BORRAR

Nota: Los datos marcados con [*] deben ser rellenados obligatoriamente.

INSERTAR NUEVA ENTRADA

Form fields:

- Nombre:*
- Pais:*
- Provincia:*
- Sector:
- Km:*
- Km Esquiabiles:
- Pistas:*
- Pistas Abiertas:
- SnowPark:
- Cotas:

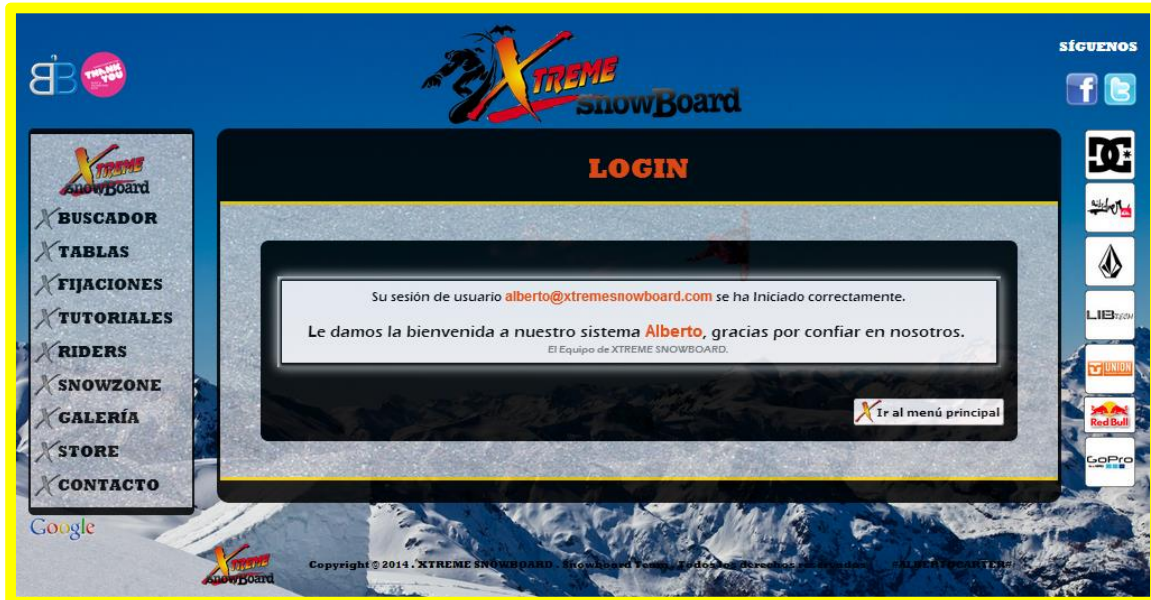
Buttons: INSERTAR, BORRAR

Nota: Los datos marcados con [*] deben ser rellenados obligatoriamente.

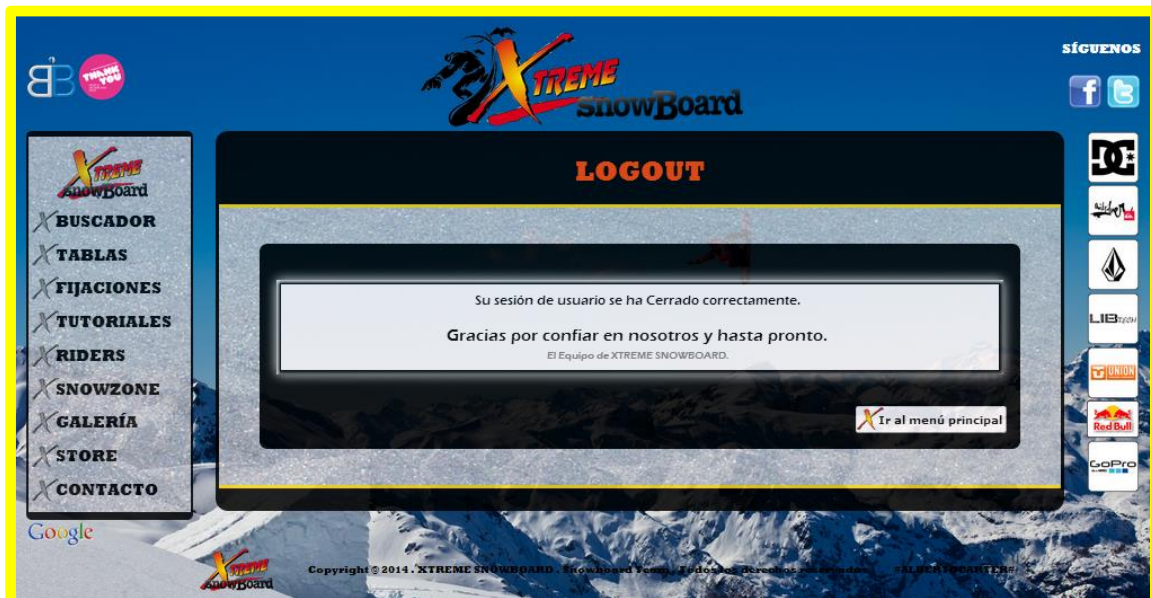
The interface includes a sidebar menu with options: ESTACIÓN, HOTEL, HABITACIÓN, CLIENTE, and SALIR. The top navigation bar shows 'Cerrar sesión' and 'Alberto Fuentes'. The right sidebar contains social media icons and various brand logos.

LOGIN.PHP*

Si el usuario se identifica correctamente le dará la bienvenida al sistema, dándole la opción de ir a su página principal de nivel **superior**. (Esta parte tiene un aspecto relevante de codificación de seguridad) Véase **Codificación de Inicio de sesión**.

**LOGOUT.PHP***

Si el usuario cierra sesión, saldrá del sistema permitiéndole así respaldar todo tipo de dato vinculante personal, manteniendo su seguridad e integridad. Se le da la opción de poder volver a su página principal de nivel **inferior**.



REGISTRO.PHP*

Aquí el usuario podrá darse de alta, previa seguridad, para poder convertirse en usuario estándar o administrador. Si introduce los datos correctamente se le podrán otorgar los permisos correspondientes.

(Esta parte tiene un aspecto relevante de codificación de seguridad)

Véase **Codificación de Registro**.

REGISTRARSE

Nombre:

1er Apellido:

2do Apellido:

NIF/PASS:

Fecha de Nac.: 1989-12-21

Teléfono:

E-Mail:

Confirmar E-Mail:

Contraseña:

Confirmar Contraseña:

Copyright © 2014. XTREME SNOWBOARD. SnowBoard Team. Todos los derechos reservados #ALBERTOCARTER#

FORMATO DE DISEÑO DE DIFERENTES APARTADOS PÚBLICOS

Si navegamos por el menú situado en la parte izquierda podemos acceder a los diferentes apartados de la web.

FORMATO: [RIDERS.PHP](#), [SNOWZONE.PHP](#), [GALERÍA.PHP](#)

Dependiendo del tipo de datos que contenga el apartado tendrá un formato de diseño diferente. Es decir, si contiene imágenes o videos se ofrece el siguiente patrón de diseño para estructurar el contenido:

The screenshot shows the website interface for 'RIDERS XTREAM SNOWBOARD'. On the left, a navigation menu lists: BUSCADOR, TABLAS, FIJACIONES, TUTORIALES, RIDERS, SNOWZONE, GALERÍA, STORE, and CONTACTO. The main content area is titled 'RIDERS XTREAM SNOWBOARD' and features a profile for 'ALBERTO FUENTES'. It includes a photo of a snowboarder in a blue and green jacket, a video player showing a snowboarder in action, and a list of details:

- Fundador del equipo XTREME SNOWBOARD
- Nacido el 20 de JUN de 1989 en Madrid (ESPAÑA)
- Modalidad: FREESTYLE
- Rider Español consolidado
- Promotor de eventos de snowboard por España y Andorra

The right sidebar contains logos for various brands including DC, Red Bull, GoPro, Oakley, Electric, Element, Vans, Special Blend, and others. The top right corner has links for 'Regístrate' and 'sÍ FUENTES' along with social media icons for Facebook and Twitter.

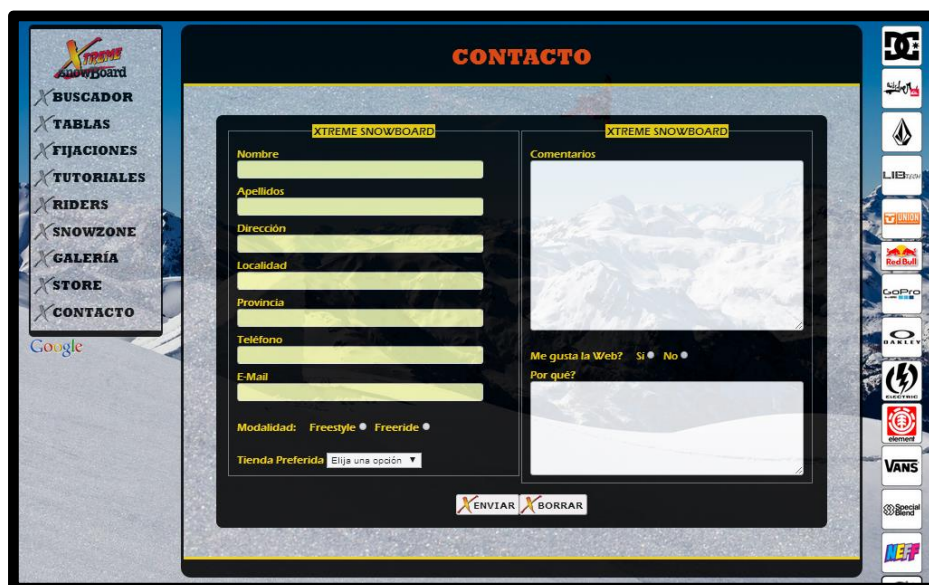
FORMATO: TABLAS.PHP, FIJACIONES.PHP, TUTORIALES.PHP, STORE.PHP

Si el contenido principal contiene texto y tablas de texto con imágenes se visualizará el siguiente formato de diseño:



FORMATO: CONTACTO.PHP

Diseño del formulario de contacto para el usuario invitado y estándar con el que podrán comunicarse vía e-mail con el administrador.



FORMATO DE DISEÑO DEL SERVICIO: BUSCADOR DE HOTELES

BUSCARHOTEL.PHP*

El usuario podrá configurar su búsqueda para iniciar el servicio de reserva de hoteles. Se le da un aspecto atractivo y desafiante, incitando así al usuario a que pruebe suerte en su búsqueda.

LISTARHOTEL.PHP*

Aquí se mostrará la búsqueda de los diferentes hoteles disponibles encontrados en la base de datos. Se da opción a que el usuario pueda reservar cada hotel visualizado. *(Esta parte tiene un aspecto relevante en la codificación del buscador)*
Véase **Codificación** del buscador de hoteles.

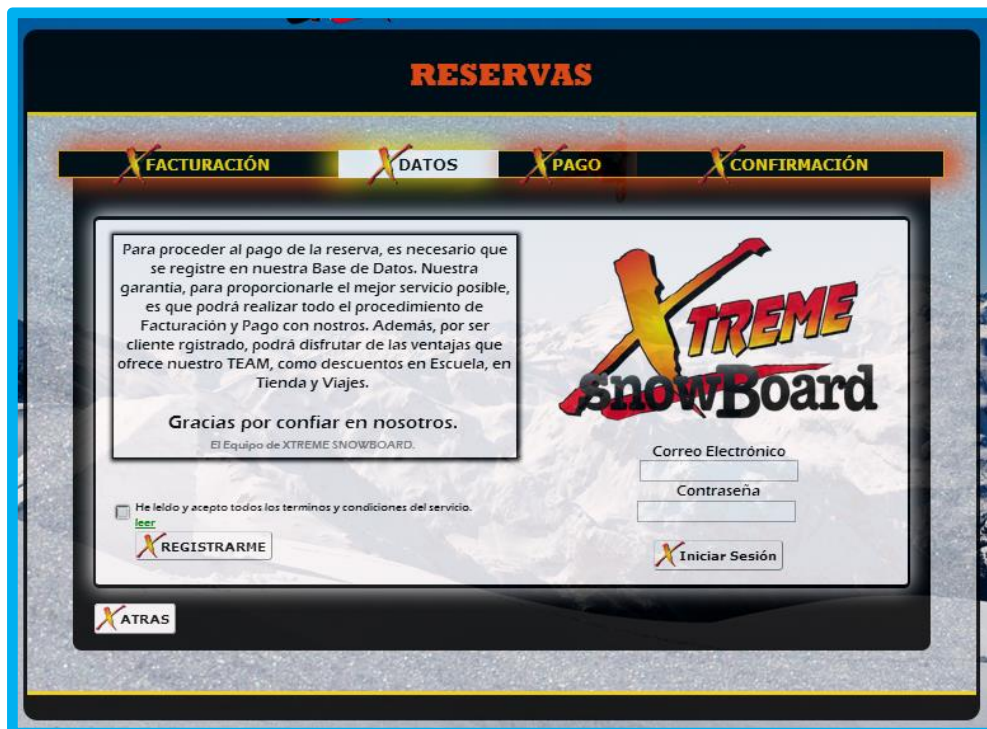
DISPONIBILIDAD DE HOTEL			
Estación: GrandValira	Soldeu	Andorra	Personas: 3
Aparta-Hotel: Cacique			
Transporte: NO	Cafetería/Restaurante: SI	Parking: NO	Internet/Wifi: SI
OFERTA: 30 €/día		Precio Total: 120€	X RESERVAR
Estación: GrandValira	Soldeu	Andorra	Personas: 3
Aparta-Hotel: DeuSol			
Transporte: SI	Cafetería/Restaurante: SI	Parking: NO	Internet/Wifi: NO
OFERTA: 60 €/día		Precio Total: 240€	X RESERVAR

FACTURACION.PHP*

Seguidamente se mostrará el presupuesto de la reserva, invitando a que el usuario confirme, o vuelva atrás para continuar con la búsqueda. Si confirma pasará al proceso de identificación del sistema.

**CLIENTE.PHP***

EL cliente del servicio ha aceptado el presupuesto, luego deberá darse de alta como nuevo usuario o tendrá que *loguearse* si ya se tiene cuenta de usuario estándar en el sistema. (Esta parte tiene un aspecto relevante en la codificación del buscador y en la de seguridad) Véase **Codificación del buscador de hoteles** y **Codificación de Inicio de sesión**.



DATOS.PHP*

Aquí el usuario podrá darse de alta, previa seguridad, para poder convertirse en usuario estándar o administrador. Si introduce los datos correctamente se le podrán otorgar los permisos correspondientes y se le dará acceso directo al pago de la reserva. *(Esta parte tiene un aspecto relevante de codificación de seguridad)*

Véase **Codificación de Registro**.

The screenshot shows a web interface titled 'RESERVAS' with a navigation bar containing 'FACTURACIÓN', 'DATOS', 'PAGO', and 'CONFIRMACIÓN'. The 'DATOS' tab is active, displaying a registration form with the following fields:

- Nombre:*
- 1er Apellido:*
- 2do Apellido:*
- NIF/PASS:*
- Fecha de Nac.:* 1989-12-21
- Teléfono:*
- E-Mail:*
- Confirmar E-Mail:*
- Contraseña:*
- Confirmar Contraseña:*

At the bottom of the form, there are two buttons: 'X ATRAS' on the left and 'X CONFIRMAR' on the right.

PAGO USUARIO.PHP* Y PAGO.PHP*

El sistema de pago a emplearse será a través de tarjeta de crédito. Se validará dicha tarjeta y se comprobará si es real o no. No habrá posibilidad de fraude. *(Esta parte tiene un aspecto relevante de codificación de seguridad)* Véase **Codificación de Pago**.

The screenshot shows the 'RESERVAS' system with the 'PAGO' tab selected in the navigation bar. The form is titled 'Por favor introduzca su información de pago' and contains the following fields:

- Tipo de Tarjeta: Master Card (dropdown menu)
- Nº de Tarjeta: 4030512785985011
- Caducidad: Febrero 2017 (dropdown menus)
- CVV: 051

At the bottom of the form, there is a button labeled 'X Proceder al PAGO'.

CONFIRMACION.PHP*

Aquí el usuario podrá visualizar su factura final. Si confirma el pago la reserva se guardará automáticamente en la base de datos, quedando la habitación de hotel inaccesible para cualquier otro usuario en la fecha configurada en la reserva. En el caso de que se abandone o se cierre la página se perdería todo el proceso y la reserva no se efectuaría.

RESERVAS

FACTURACIÓN | DATOS | PAGO | CONFIRMACIÓN

XTREME SnowBoard Fecha: 2014-05-27 N° de Factura: 9 XTREME SNOWBOARD España S.A. Edificio Xtreme Snowboard * C/Finestrat, 21 Las Rozas * 28290 * Madrid * España

CLIENTE		PAGO	
Nombre: Alberto Apellidos: Fuentes DNI/NIE: 12345678Z Telefono: 636859678 Email: albertocarter@xtremesnowboard.com		Sistema: Tarjeta bancaria Tipo: VISA N° Tarjeta: xxxx-xxxx-xxxx-5011	

ID	HOTEL	ESTACION	HABITACION	PRECIO	DÍAS DE ESTANCIA
3	Aparta-Hotel Deusol	GrandValira	3 pers.	60 €/Dia	4

PRECIO A PAGAR: **240 €**

[Cambiar Pago](#) [PAGAR](#)

FINALIZACION.PHP*

Finalmente se le muestra al cliente una copia de su recibo y se lanza un mensaje de confirmación de reserva. Además se da la opción de volver al menú principal de usuario.

RESERVAS

Reserva realizada con éxito

Enhorabuena Alberto:

El proceso de Pago y Reserva se ha realizado correctamente. Ahora podrá disfrutar de su reserva de Hotel de la mano de Xtreme Snowboard. Se ha enviado una copia de la tramitación de Pago y Reserva a su correo electrónico. Deberá entregar dicha documentación impresa a la hora de efectuar su entrada en el Hotel. Para cualquier duda, sugerencia o anulación de reserva, recuerde que estamos a su disposición las 24 horas.

atencion.cliente@xtremesnowboard.com

Gracias por confiar en nosotros y buen viaje.
El Equipo de XTREME SNOWBOARD.

Nº FACTURA	HOTEL	ESTACION	HABITACION	ENTRADA	SALIDA
9	Aparta-Hotel Deusol	GrandValira	3 pers.	2014-05-20	2014-05-24

PAGADO: **240 €**

[Ir al menú principal](#)

4. CODIFICACIÓN

ENTORNO DE PROGRAMACION

Proyecto desarrollado en **Google Chrome**. Para su visualización más óptima se recomienda utilizar dicho navegador, pues se ha implantado código **HTML5** y **CSS3**. En caso de ejecutar la web en un navegador diferente, no se garantiza su visualización más óptima.

La web también ha sido desarrollada para ajustarse a las resoluciones de *smartphones* y *tablets*.

LENGUAJES Y HERRAMIENTAS

Los lenguajes de codificación utilizados son los siguientes:

PHP: para el control de las bases de datos, las comprobaciones o validaciones de seguridad, y las consultas y operaciones referentes a los datos alojados en ella.

HTML: para el diseño estructural de la web.

CSS: para los estilos de diseño de la web.

JAVASCRIPT: para la interacción del sistema y validaciones de sistema de pago.

La herramienta de software utilizada para desarrollar el código ha sido **Sublime Text 3**, uno de los editores de código más profesionales que hay en la actualidad para los programadores y desarrolladores.

Se ha utilizado la herramienta de programa **WAMP Server** como escenario de edición y de pruebas. Éste ofrece una instalación rápida de un servidor web local en nuestra máquina para poder almacenar las bases de datos y poder ejecutar el código para visualizarlo en un navegador. De esta forma podemos trabajar rápidamente si tenemos instalado el software en cualquier host donde vallamos a trabajar.

ASPECTOS RELEVANTES DE LA IMPLEMENTACIÓN

CODIFICACIÓN DE REGISTRO

A continuación se detalla el proceso de validación implantado para la comprobación de datos del registro del usuario en el sistema. Al final de la explicación se mostrará el código.

REGISTRO.PHP* Y DATOS.PHP*

Paso 1

Si confirmamos el formulario sin introducir ningún dato, se comprobará lo siguiente:

Paso 2

Nombre:*

1er Apellido:*

2do Apellido:

NIF/PASS:*

Fecha de Nac.:*

Teléfono:*

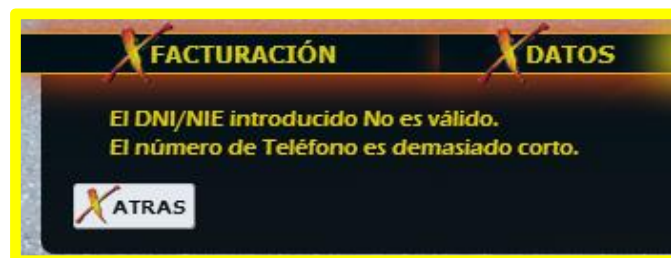
E-Mail:*

Confirmar E-Mail:*

Contraseña:*

Confirmar Contraseña:*

Al introducir los primeros campos obligatorios, Nombre y Apellido, se pasara a comprobar el DNI y el teléfono. Si introducimos un formato de DNI o teléfono erróneo, se comprobará. Si además el teléfono no es de tipo numérico o tiene menos de 9 caracteres se alertará al usuario.



Paso 3

Nombre:*

1er Apellido:*

2do Apellido:

NIF/PASS:*

Fecha de Nac.:*

Teléfono:*

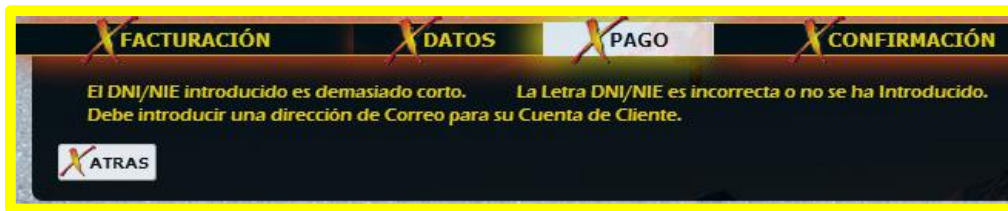
E-Mail:*

Confirmar E-Mail:*

Contraseña:*

Confirmar Contraseña:*

Si introducimos un DNI con menos de 8 números se alertará al usuario, además se recuerda que debe introducir la letra identificativa. En la siguiente comprobación se le pide al usuario que introduzca una dirección de correo para su cuenta de cliente.



Paso 4

Nombre:* Alberto

1er Apellido:* Fuentes

2do Apellido: *

NIF/PASS:* 05304958

Fecha de Nac.:* 1989-12-21

Teléfono:* 636859678

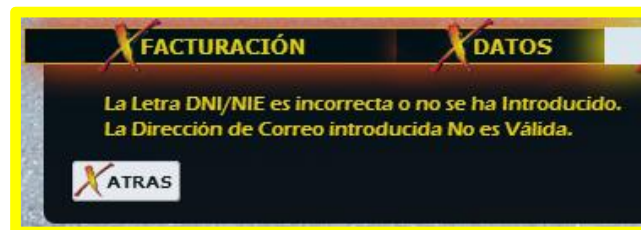
E-Mail:* alberto

Confirmar E-Mail: *

Contraseña: *

Confirmar Contraseña: *

A continuación introducimos los 8 números del DNI, si no introducimos la letra o la letra es incorrecta se alertará al usuario. Si intentamos introducir un formato de dirección de correo no válido avisaremos del error.



Paso 5

Nombre:* Alberto

1er Apellido:* Fuentes

2do Apellido: *

NIF/PASS:* 12345678Z

Fecha de Nac.:* 1989-12-21

Teléfono:* 636859678

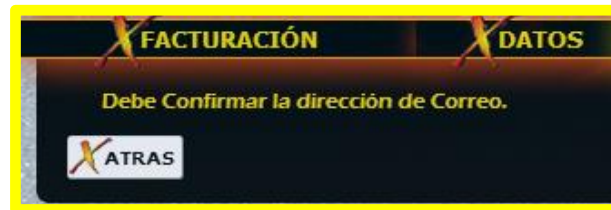
E-Mail:* albertocarter@xtremesnc

Confirmar E-Mail: *

Contraseña: *

Confirmar Contraseña: *

Si introducimos un formato de correo electrónico correcto y no confirmamos la dirección de correo, se mostrará el siguiente mensaje.



Paso 6

Nombre:*	Alberto
1er Apellido:*	Fuentes
2do Apellido:	
NIF/PASS:*	05304958P
Fecha de Nac.:*	1989-12-21
Teléfono:*	636859678
E-Mail:*	alberto@xtremcsnowboa
Confirmar E-Mail:*	alberto@xtremcsnowboa
Contraseña:*	
Confirmar Contraseña:*	

Si el DNI introducido es correcto y el correo electrónico ha sido confirmado y validado, sólo en el caso de que ya exista un usuario en el sistema con el DNI o correo introducido, se mostrarán los siguientes mensajes.



Paso 7

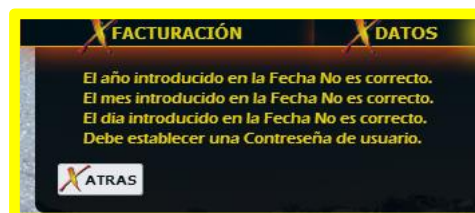
Nombre:*	Alberto
1er Apellido:*	Fuentes
2do Apellido:	
NIF/PASS:*	12345678Z
Fecha de Nac.:*	1989/12
Teléfono:*	636859678
E-Mail:*	albertocarter@xtremcsnc
Confirmar E-Mail:*	albertocarter@xtremcsnc
Contraseña:*	
Confirmar Contraseña:*	

Al introducir un formato de fecha distinto de “aaaa-mm-dd” o una fecha incompleta se muestra el siguiente mensaje para el usuario. También se comunica que hay que establecer una contraseña como pasa final.



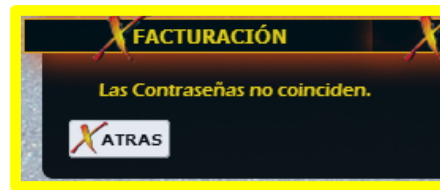
Paso 8

Si el usuario introduce un formato de fecha válido pero no real, se muestran los siguientes mensajes dependiendo del año, mes o día introducidos.



Paso 9

En el caso de que la confirmación de contraseña no coincida con la contraseña, se alertará al usuario con el siguiente mensaje.



CÓDIGO FUENTE

```
//Instrucciones:
$nombre=$REQUEST['nombre'];
$apellido_uno=$REQUEST['apellido_uno'];
$apellido_dos=$REQUEST['apellido_dos'];
$nif_pass=$REQUEST['nif_pass'];
$fech_nac=$REQUEST['fech_nac'];
$telefono=$REQUEST['telefono'];
$email=$REQUEST['email'];
$conf_email=$REQUEST['conf_email'];
$clave=$REQUEST['clave'];
$conf_clave=$REQUEST['conf_clave'];

//Encriptamos La Contraseña:
$password=sha1($clave);

//Codificamos La variable clave:
$clave = base64_decode($clave);
$conf_clave = base64_decode($conf_clave);

//Consulta de Email y DNI/NIE existentes en La BBDD:
$instr_comprobar_email = "select * from cliente where Email='$email'";
$consulta_comprobar_email = mysql_query($instr_comprobar_email, $conexion) or die ("Fallo en la comprobación de Email NIF existentes.");
$box1 = mysql_fetch_array($consulta_comprobar_email);

$email_X = $box1['Email'];

$instr_comprobar_nif_pass = "select * from cliente where NIF_PASS='$nif_pass'";
$consulta_comprobar_nif_pass = mysql_query($instr_comprobar_nif_pass, $conexion) or die ("Fallo en la comprobación de Email NIF existentes.");
$box2 = mysql_fetch_array($consulta_comprobar_nif_pass);

$nif_passX = $box2['NIF_PASS'];

//COMPROBACIONES
$boton_atras=1;
//Comprobación campos obligatorios:
if ((trim($nombre)=="") || (trim($apellido_uno)=="")) {
    print("<span style='margin-left: 5%;>Debe rellenar todos los campos obligatorios (<span style='color: red;*></span>).</span><br>");
}

//Comprobación DNI/NIE:
if ($nif_pass=="") {
    print("<span style='margin-left: 5%;>Debe introducir su DNI/NIE para registrar su Cuenta de Cliente.</span><br>");
}
elseif ($nif_pass!="") {
    if (!is_numeric(substr($nif_pass, 0, 8))) {
        print("<span style='margin-left: 5%;>El DNI/NIE introducido No es válido.</span><br>");
    }
    else{
        if(strlen($nif_pass)<8) {
            print("<span style='margin-left: 5%;>El DNI/NIE introducido es demasiado corto.</span>");
        }

        $nif_pass = strtoupper($nif_pass);

        $letra = substr($nif_pass, -1, 1);
        $numero = substr($nif_pass, 0, 8);

        // Si es un NIE hay que cambiar La primera Letra por 0, 1 ó 2 dependiendo de si es X, Y o Z.
        $numero = str_replace(array('X', 'Y', 'Z'), array(0, 1, 2), $numero);

        $modulo = $numero % 23;
        $letras_validas = "TRWAGMYFPDXBNJZSQVHLCKE";
        $letra_correcta = substr($letras_validas, $modulo, 1);

        if($letra_correcta==$letra) {
            print("<span style='margin-left: 5%;>La Letra DNI/NIE es incorrecta o no se ha Introducido.</span><br>");
            //echo "$letra_correcta";
        }
    }
}
if (trim($nif_passX)!="") {
    if ($nif_pass==$nif_passX) {
        print("<span style='margin-left: 5%;>Ya Existe un Cliente con el DNI/NIE especificado en nuestra base de datos.</span><br>");
    }
}
}
```

```

//Comprobación Formato Fecha:
if (!preg_match('/^\d{4}\-\d{1,2}\-\d{1,2}$/i', $fech_nac)) {
    print("<span style='margin-left: 5%;>El Formato de Fecha introducido No es Válido.</span><br>");
}
if (preg_match('/^\d{4}\-\d{1,2}\-\d{1,2}$/i', $fech_nac)) {
    list($ano,$mes,$dia) = explode('-', $fech_nac);

    if ($ano > 2014) {
        print("<span style='margin-left: 5%;>El año introducido en la Fecha No es correcto.</span><br>");
    }
    if ($mes > 12) {
        print("<span style='margin-left: 5%;>El mes introducido en la Fecha No es correcto.</span><br>");
    }
    if ($dia > 31) {
        print("<span style='margin-left: 5%;>El día introducido en la Fecha No es correcto.</span><br>");
    }
}

//Comprobación Telefono:
if (!is_numeric($telefono)) {
    print("<span style='margin-left: 5%;>Debe introducir un numero de Teléfono válido.</span><br>");
}
elseif (strlen($telefono) < 9) {
    print("<span style='margin-left: 5%;>El número de Teléfono es demasiado corto.</span><br>");
}

//Comprobación Email:
elseif ((trim($email) == "")) {
    print("<span style='margin-left: 5%;>Debe introducir una dirección de Correo para su Cuenta de Cliente.</span><br>");
}
elseif (!preg_match('/^[A-Za-z0-9_+]+@[A-Za-z0-9-]+\.[A-Za-z]{2,4}$/i', $email)) {
    print("<span style='margin-left: 5%;>La Dirección de Correo introducida No es Válida.</span><br>");
}
elseif ((trim($conf_email) == "")) {
    print("<span style='margin-left: 5%;>Debe Confirmar la dirección de Correo.</span><br>");
}
elseif ($email != $conf_email) {
    print("<span style='margin-left: 5%;>Las direcciones de Correo no coinciden.</span><br>");
}
elseif (trim($email_X) != "") {
    if ($email == $email_X) {
        print("<span style='margin-left: 5%;>Ya Existe la Dirección de Correo especificada en nuestra base de datos.</span><br>");
    }
}
}

//Comprobación contraseña:
elseif ((trim($clave) == "")) {
    print("<span style='margin-left: 5%;>Debe establecer una Contraseña de usuario.</span><br>");
}
elseif ((trim($conf_clave) == "")) {
    print("<span style='margin-left: 5%;>Debe Confirmar la Contraseña de usuario.</span><br>");
}
elseif (strlen($clave) > 30) {
    print("<span style='margin-left: 5%;>La Contraseña no puede contener mas de 30 Caracteres.</span><br>");
}
elseif ($clave != $conf_clave) {
    print("<span style='margin-left: 5%;>Las Contraseñas no coinciden.</span><br>");
}
}

else{
    $boton_atras=0;
    $instr_insertar = "insert into cliente(Nombre,1Apellido,2Apellido,NIF_PASS,Fech_Nac,Telefono,Email,Password)
    values('$nombre', '$apellido_uno', '$apellido_dos', '$nif_pass', '$fech_nac', '$telefono', '$email', '$password');

    //Consulta:
    $consulta_insertar = mysql_query($instr_insertar, $conexion)
    or die ("<br><span style='margin-left: 5%;>ERROR: Email o NIF duplicados<div style='width: 95%; margin-left: 21px;' ID='Principal_Form_Btn'>
    <A HREF='javascript:history.back()'><input type='submit' value='ATRAS' ID='Form_Btn'></A></div>");
    mysql_close($conexion);

    print("<span style='margin-left: 20px;>Bienvenido/a $nombre.</span>");
    print("<center><br><div ID='Principal_Box_Reserva' style='border: outset'>");
    ?>

<DIV ID="Alta_NFW_USER">

```

CODIFICACIÓN DE INICIO DE SESIÓN

A continuación se detalla el proceso de validación implantado para el inicio de sesión del usuario en el sistema. Al final de la explicación se mostrará el código.

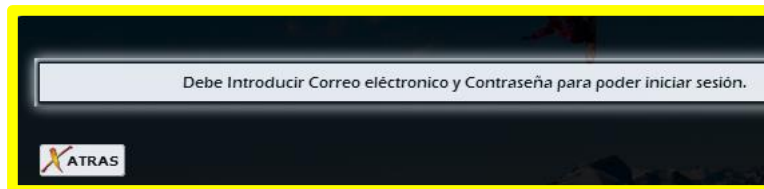
LOGIN.PHP* Y LOGIN PAGO.PHP*

Paso 1



The screenshot shows the login interface for 'XTREME SnowBoard'. At the top, there is a logo with a snowboard and the text 'XTREME SnowBoard'. Below the logo, there are two input fields: 'Correo Electrónico' and 'Contraseña'. Underneath the 'Contraseña' field is a button labeled 'Iniciar Sesión' with a snowboard icon. Below the button is a red link labeled 'Regístrate'. The entire form is enclosed in a yellow border.

Si se inicia sesión sin introducir ningún dato, se muestra el siguiente mensaje.



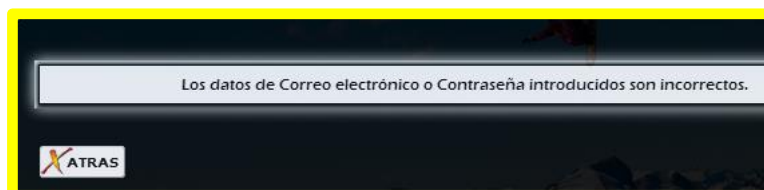
The screenshot shows a dark grey error message box with a white border. The text inside reads: 'Debe Introducir Correo electrónico y Contraseña para poder iniciar sesión.' In the bottom left corner, there is a small logo with the text 'ATRAS'.

Paso 2



The screenshot shows the same login interface as in Step 1, but the 'Correo Electrónico' field is filled with a series of dots, and the 'Contraseña' field is filled with a series of asterisks. The 'Iniciar Sesión' button and 'Regístrate' link are still visible below the fields.

Si se introduce un correo electrónico que no está en la base de datos del sistema, o se introduce mal la contraseña de usuario, se lanzará el siguiente mensaje.



The screenshot shows a dark grey error message box with a white border. The text inside reads: 'Los datos de Correo electrónico o Contraseña introducidos son incorrectos.' In the bottom left corner, there is a small logo with the text 'ATRAS'.

CODIFICACIÓN DE SESIÓN

A continuación se detalla el proceso que realiza la cabecera de sesión de cada una de las páginas del portal web. Al final de la explicación se mostrará el código.

TODAS LAS PÁGINAS

En cada una de las páginas del proyecto web, exceptuando la parte de registro o inicio de sesión, podemos observar el nombre del usuario identificado y la opción de poder cerrar la sesión en cualquier momento.



En el caso de que el usuario sea administrador, en la cabecera de sesión aparecerá un botón que dará acceso a la interfaz web de administración de las bases de datos del sistema.



CÓDIGO FUENTE

```
<?php
//Iniciar sesión o Mantener sesión:
session_start();
?>
<HTML>
<HEAD>
<TITLE>Xtreme Snowboard</TITLE>
<link rel="stylesheet" type="text/css" href="xtremesnowboard.css" />
</HEAD>
<BODY>
<DIV ID="CONTENEDORA">
<?php
print("<div style='color: gold; font-family: Arial; font-weight: bold;'>");
if (isset($_SESSION["admin"])) {
    print("<a href='gestion_estacion.php' style='color: snow; font-weight: normal;'>
    <input type='button' name='admin' value='ADMINISTRACION' ID='Form_Btn' style='text-align: left;' /></a>");
}
if (isset($_SESSION["usuario_valido"])){
    print("<input type='button' name='encajar' value='' style='text-align: left; visibility: hidden;' />");
    print("<span style='float: right;'>");
    print($_SESSION["usuario_valido"]);
    print("</span>");
    print("<a href='logout.php' style='float: right; color: snow; font-weight: normal; margin-right: 15px;'>Cerrar sesión</a>");
}
else{
    print("<input type='button' name='encajar' value='' style='text-align: left; visibility: hidden;' />");
    print("<a href='registro.php' style='float: right; color: gold;'>Regístrate</a>");
}
print("</div>");
?>
</DIV ID="CABECERA">
```

CODIFICACIÓN DEL BUSCADOR DE HOTELES

A continuación se detalla el código de los procesos que realiza el servicio de buscador de hoteles.

LISTARHOTEL.PHP*

El buscador de hoteles realiza una búsqueda recursiva en las bases de datos del sistema según la configuración anteriormente seleccionada por el usuario.



CÓDIGO FUENTE

```
function dias_transcurridos($fecha_i,$fecha_f)
{
    $Dias = (strtotime($fecha_i)-strtotime($fecha_f))/86400;
    $Dias = abs($Dias); $Dias = floor($Dias);
    return $Dias;
}
}

<?php
// Buscar Hotel en la base de datos:
if (isset($_REQUEST['buscar_aloj'])) {

    // Conectar con la base de datos:
    $conexion = mysql_connect("127.0.0.1","root","137280") or die ("No se puede conectar con el Servidor.");

    // Seleccionamos la base de datos:
    mysql_select_db("xtremesnowboard") or die ("No se puede seleccionar la base de datos.");

    //Instrucciones:
    $estacion=$_REQUEST['estacion'];
    $num_personas=$_REQUEST['num_personas'];
    $entrada=$_REQUEST['entrada'];
    $salida=$_REQUEST['salida'];

    //Buscar Hotel:
    $instr_vista_disponible = "create view disponible as(select distinct A.ID_ALOJ, A.Nombre, H.Num_Personas, A.Tipo, A.Direccion, E.Nombre as 'Estacion'
    from estacion E, est_contiene_aloj X, alojamiento A, habitacion H where (E.ID_EST=X.ID_EST AND X.ID_ALOJ=A.ID_ALOJ AND A.ID_ALOJ=H.ID_Aloj)
    AND E.ID_EST='estacion' AND H.Num_Personas='num_personas' AND H.Disponible like 'SI');";

    $consulta_vista_disponible = mysql_query($instr_vista_disponible, $conexion) or die ("Fallo en la creacion de la Vista Disponible.");

    $instr_vista_reservado = "create view reservado as(select distinct A.ID_ALOJ, A.Nombre, H.Num_Personas, A.Tipo, A.Direccion, E.Nombre as 'Estacion'
    from estacion E, est_contiene_aloj X, alojamiento A, client_reserva_aloj R, habitacion H
    where (E.ID_EST=X.ID_EST AND X.ID_ALOJ=A.ID_ALOJ AND A.ID_ALOJ=R.ID_Aloj AND R.Num_Personas=H.Num_Personas)
    AND E.ID_EST='estacion' AND H.Num_Personas='num_personas' AND R.Fecha_Entrada BETWEEN '$entrada' and '$salida');";

    $consulta_vista_reservado = mysql_query($instr_vista_reservado, $conexion) or die ("Fallo en la creacion de la Vista Reservado.");

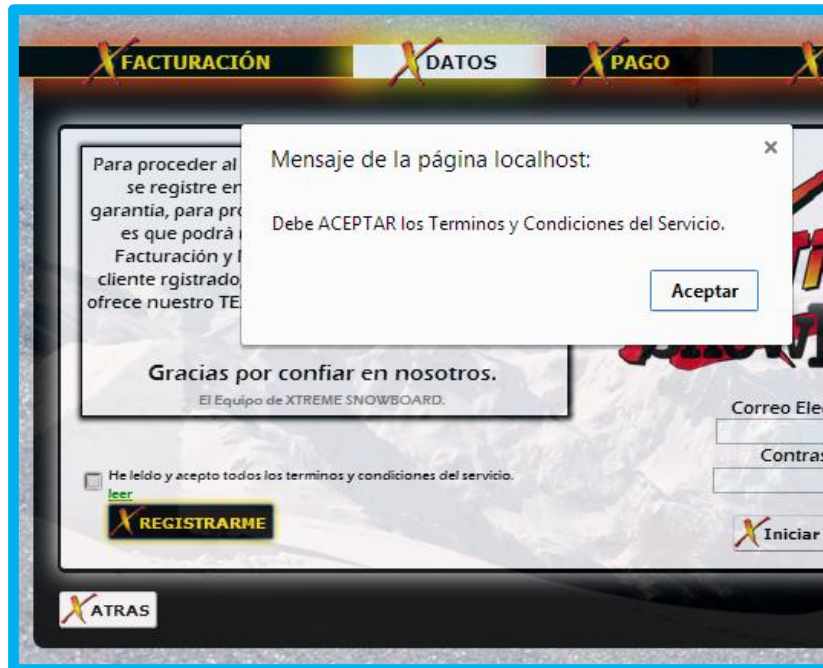
    $instr_busca = "select * from disponible where (ID_ALOJ, Num_Personas) NOT IN (Select ID_ALOJ,Num_Personas from reservado)";

    if (isset($_POST['transporte']) or isset($_POST['restaurante']) or isset($_POST['parking']) or isset($_POST['wifi'])) {

        if (isset($_POST['transporte'])) {
            $instr_busca=$instr_busca." and Transporte='SI'";
        }
        if (isset($_POST['restaurante'])) {
            $instr_busca=$instr_busca." and Restaurante='SI'";
        }
        if (isset($_POST['parking'])) {
            $instr_busca=$instr_busca." and Parking='SI'";
        }
        if (isset($_POST['wifi'])) {
            $instr_busca=$instr_busca." and Wifi='SI'";
        }
    }
}
```


CLIENTE.PHP*

Para acceder a registrarse como usuario del sistema en el proceso de pago de la reserva de hotel, es necesario marcar que se han aceptado los términos y condiciones del servicio para poder continuar. Si hacemos clic en [leer](#) podremos acceder a la documentación del servicio en formato PDF.

**CÓDIGO FUENTE**

```

<script type="text/javascript">
function estado(){
  if (document.getElementById('acepto').checked) {
    document.getElementById('envio').submit();
  }
  else{
    alert("Debe ACEPTAR los Terminos y Condiciones del Servicio.");
    document.getElementById('acepto').focus();
    return false;
  }
}
</script>

<td><?php
  print("<a href='datos.php?q=$entrada&r=$salida&s=$Precio&t=$Nombre&u=$Tipo&v=$Estacion&w=$Dias&x=$Pagar&y=$ID_ALOJ&z=$Num_Personas' id='envio'>
  <input type='submit' value='REGISTRARME' ID='Form_btn' onClick='javascript:RetornarEstado()' /></a>");
?>
</td>

```


ccvalidate.js

```
(function($) {
  $.fn.extend({
    ccvalidate: function(options) {
      var settings = $.extend({
        onvalidate: null
      }, options);
      var cards = new Array();
      cards[0] = { name: "mcd", clength: [16], exp: /^[1-5]/ };
      cards[1] = { name: "vis", clength: [13, 16], exp: /^4/ };
      cards[2] = { name: "amx", clength: [15], exp: /^3(4|7)/ };
      cards[3] = { name: "dnr", clength: [14], exp: /^3[0,6,8]\d{12}/ };
      cards[4] = { name: "dis", clength: [16], exp: /^6011\d{4}\d{4}\d{4}/ };
      return this.each(function() {
        var type = $(this).find('.cc-ddl-type');
        var cardnumber = $(this).find('.cc-card-number');
        var checkout = $(this).find('.cc-checkout');

        if (type == null || type == undefined)
          alert('Type Drop Down List not supplied');
        if (cardnumber == null || cardnumber == undefined)
          alert('Card Number Text Box not supplied');
        if (checkout == null || checkout == undefined)
          alert('Checkout button not supplied');

        if (type != null && type != undefined &&
            cardnumber != null && cardnumber != undefined &&
            checkout != null && checkout != undefined) {

          $(checkout).click(function() {
            return ValidateCreditCard(cardnumber.val(), type.val());
          });
        }
      });
    }
  });
});
```

```
function ValidateCreditCard(cardNumber, cardType) {
  var isValid = false;
  var ccCheckRegExp = /^[^d ]/;
  isValid = !ccCheckRegExp.test(cardNumber);
  if (isValid) {
    var cardNumbersOnly = cardNumber.replace(/ /g, ""); //trim spaces
    var cardNumberLength = cardNumbersOnly.length;
    var lengthIsValid = false;
    var prefixIsValid = false;
    var prefixRegExp;
    var card = getCardType(cardType);
    if (card != null) {
      for (i = 0; i < card.clength.length; i++) {
        if (!lengthIsValid) {
          if (cardNumberLength == card.clength[i]) lengthIsValid = true;
        }
      }
      prefixIsValid = card.exp.test(cardNumbersOnly);
    }
    isValid = prefixIsValid && lengthIsValid;
  }
  if (isValid) {
    var checkSumTotal = 0;
    checkSumTotal = computeChecksum(cardNumbersOnly);
    isValid = (checkSumTotal % 10 == 0);
  }
  if (settings.onvalidate != null)
    settings.onvalidate(isValid);
  return isValid;
}

function getCardType(type) {
  var card = null;
  for (i = 0; i < cards.length; i++) {
    if (cards[i].name.toLowerCase() == type.toLowerCase()) {
      card = cards[i];
      break;
    }
  }
  return card;
}

function computeChecksum(cardNo) {
  var checksum = 0;
  var factor = 1;
  var temp;
  for (i = cardNo.length - 1; i >= 0; i--) {
    temp = Number(cardNo.charAt(i)) * factor;
    if (temp > 9) {
      checksum += 1;
      temp -= 10;
    }
    checksum += temp;
    factor = (factor == 1 ? 2 : 1);
  }
  return checksum;
}
});
```


pago.php y pago_usuario.php

```

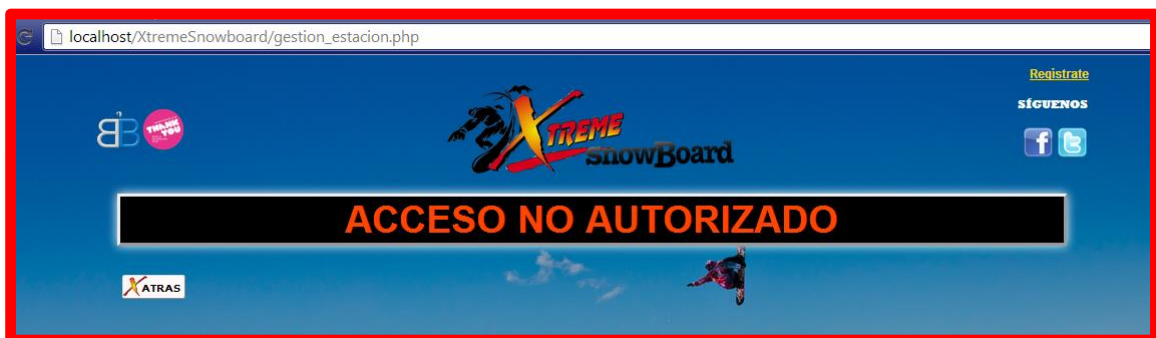
<script type="text/javascript">
$(document).ready(function() {
  $('#cc-container').ccvalidate({ onvalidate: function(isValid) {
    if (!isValid) {
      alert('La Tarjeta de Crédito No es válida.');
```

CODIFICACIÓN DE ACCESO NO AUTORIZADO

A continuación se muestra el código implantado para denegar el acceso a diferentes apartados de la web, a todo usuario ajeno o sin permisos.

GESTION (apartado).PHP*

Si intentamos acceder como usuario invitado o usuario estándar a la interfaz de administración a través de la barra de direcciones, se lanzará el siguiente mensaje.



CÓDIGO FUENTE

Incluimos desde el inicio de la página excluyendo la cabecera:

```

if (isset($_SESSION['admin']) && $_SESSION['admin']==TRUE) {
?>
<DIV ID="MENU_LEFT">
  <DIV ID="MENU_LEFT_TOP">
    <DIV ID="Menu_logo">
      <A href="index.php" target=_top><CENTER><IMG src="imagenes\logoweb2.png" width=70%></CENTER></A>
    </DIV>
  </DIV>

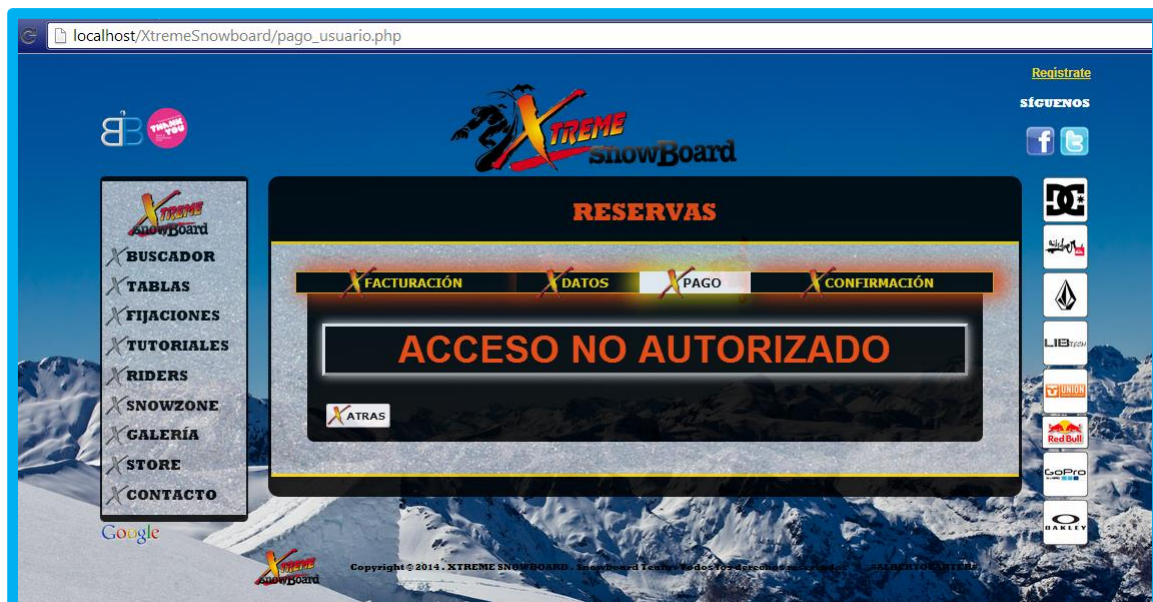
```


Hasta el final de la página:

```
<TD align=left width=20%#ALBERTOCARTER</TD>
</TABLE>
</DIV>
</DIV>
<?php
}
else{
print("<center><br><div ID='Principal_Box_Reserva' style='border: outset'><DIV
style= background-color: black; color: orangered; font-size: 50px; font-family: ARIAL; font-weight: bold;'>");
print("ACCESO NO AUTORIZADO");
print("</DIV></div><center><br>");
print("<div style= width: 95%; ID='Principal_Form_Btn'>");
print("<table style= width:100%,text-align: left; ><tr><td><A HREF='javascript:history.back()'>
<input type='submit' value='ATRÁS' ID='Form_Btn' /></A></td></tr></table></div>");
}
}
?>
</BODY>
```

PAGO.PHP* , PAGO USUARIO.PHP* , CONFIRMACION.PHP* Y FINALIZACION.PHP*

Si intentamos acceder como usuario invitado a las páginas referentes al pago del servicio de reserva de hoteles, a través de la barra de direcciones, se lanzará el siguiente mensaje.



CÓDIGO FUENTE

Se implanta el mismo código visto en el apartado anterior dentro del cuadro central de cada página.

5. PRUEBAS

Una vez probadas todas las páginas y funcionalidades de la web no se han encontrado errores de consistencia o navegación. Las pruebas se han mostrado gráficamente en el apartado de codificación. Hasta conseguir completar cada comprobación o validación se ha aplicado el método de ensayo y error para conseguir el resultado esperado.

Las pruebas han consistido en:

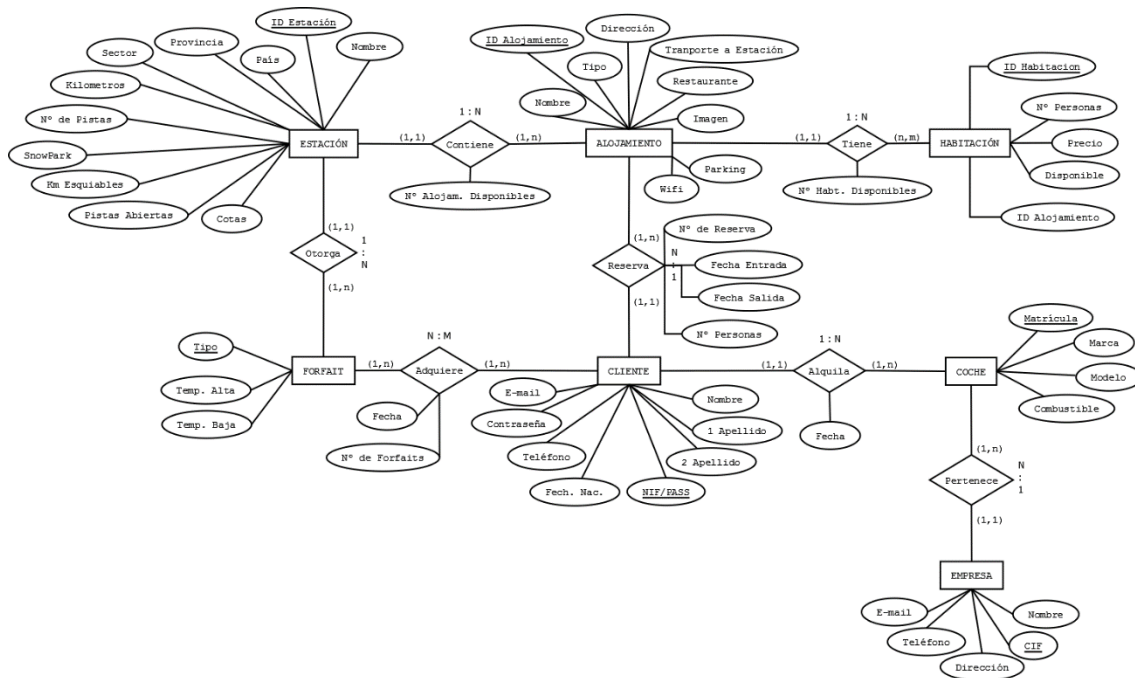
- Inserción, modificación y eliminación de entradas en la base de datos a través de la interfaz web de administración.
- Realizar reservas de hotel como usuario invitado para acceder al portal de cliente, y **registrarnos** como nuevo usuario para completar el proceso de pago y finalización.
- Realizar reservas de hotel como usuario invitado para acceder al portal de cliente, y **loguearnos** como usuario estándar o administrador para completar el proceso de pago y finalización.
- Realizar reservas de hotel como usuario estándar o administrador para evitar acceder al portal de cliente, y **pasar directamente** a completar el proceso de pago y finalización.
- Introducir datos erróneos y encontrar agujeros de seguridad en los formularios de registro para maximizar la seguridad de la web.
- Intentar acceder a la interfaz web de administración desde la barra de direcciones como usuario invitado o estándar y comprobar el **acceso no autorizado**.
- Navegar por toda la web buscando no perder el control de las sesiones de usuario.

6. CONCLUSIONES

Se ha finalizado el proyecto con todos los objetivos cubiertos de manera muy satisfactoria. Se ha establecido una estructura de usuarios y permisos para una herramienta de servicio útil como es un buscador de hoteles. El usuario estándar sólo puede utilizar los servicios proporcionados por los usuarios administradores, y el usuario administrador puede manejar todo el servicio que ofrece al usuario estándar. No existe la posibilidad de rellenar una ficha de registro con datos erróneos o de fraude aparentemente, pues la comprobación de alta de usuario es muy estricta, exigiendo como por ejemplo, un DNI/NIE real. Se establece un acceso no autorizado a todo usuario invitado o estándar a la interfaz de administración, de la misma forma denegamos el acceso al usuario invitado a páginas de proceso de pago del servicio. La interfaz es cómoda, directa y amigable, desarrollada incluso para *Smartphones* y *tablets*.

7. POSIBLES AMPLIACIONES Y MEJORAS

La información del futuro del Portal Web está **clasificada**. Podemos asegurar que existe un extenso trabajo que está en desarrollo y contamos con estructuras ya funcionales para la ampliación de bastantes más servicios. A continuación os mostramos un **pequeño adelanto** de la estructura de la base de datos que ya es totalmente funcional y operativa junto con su correspondiente interfaz **web**.



Gracias y hasta muy pronto.

Alberto Fuentes Sánchez

Administrador de Sistemas Informáticos en Red
 Programador y Diseñador Web
 Desarrollador y Diseñador Gráfico
 Administrador y Gestor de Bases de Datos en Red
 Director de Marketing Xtreme Snowboard®