



UNIVERSIDAD
POLITÉCNICA
DE MADRID

Grado en Ingeniería Informática

Desarrollo de una aplicación Web

Descripción breve

Desarrollo de una aplicación web diseñada para cubrir las necesidades de los estudiantes Erasmus

AUTOR: Luis Miranda Borrachero

TUTOR: Vicente Martínez

ÍNDICE

| | |
|---------------------------------------|----|
| Desarrollo de una aplicación Web..... | 1 |
| Introducción..... | 1 |
| Funcionalidades del servicio | 2 |
| Método de trabajo..... | 4 |
| Diseño del servicio web | 5 |
| Diseño de la Base de Datos | 5 |
| Entidades..... | 6 |
| Relaciones que generan tabla | 8 |
| Relaciones que no generan tabla..... | 9 |
| Diseño de la página web | 11 |
| PantallaInicio.php..... | 11 |
| PantallaInicioSesion.php | 12 |
| Inicio_sesion.php | 13 |
| Cerrarsesion.php..... | 13 |
| Registro_usuario.php..... | 14 |
| Input_datos.php | 14 |
| Acceso_procesos.php | 15 |
| DevolverProcesos.php..... | 15 |
| Marcar_proceso.php..... | 16 |
| Foros.php..... | 17 |
| Acceso_foros.php..... | 17 |
| DevolverComentarioF.php | 18 |
| Dar_like.php | 19 |
| Añadir_comentario.php | 20 |
| Añadir_comentarioBE.php | 20 |
| Valoraciones.php..... | 20 |
| DevolverValoracionesF.php..... | 21 |
| Añadir_valoraciones.php | 22 |
| Añadir_valoracionesBE.php | 23 |
| Acceso_chat.php..... | 23 |
| Devolver_mensajes.php..... | 24 |
| Introducir_mensaje.php..... | 25 |
| Chats_privados.php..... | 25 |

| | |
|---|----|
| Explorar_chats.php..... | 26 |
| Buscar_usuario.php..... | 27 |
| Crear_chat.php | 28 |
| Elementos comunes en los scripts | 29 |
| Inserción de datos iniciales | 31 |
| Obtención de los datos..... | 31 |
| Manejo e introducción de los datos..... | 31 |
| Test de usabilidad | 34 |
| Diseño de las acciones a realizar | 34 |
| Resultados..... | 34 |
| Usuario 1:..... | 34 |
| Usuario 2:..... | 35 |
| Usuario 3:..... | 36 |
| Usuario 4:..... | 36 |
| Usuario 5:..... | 37 |
| Análisis de los resultados. | 37 |
| Cambios realizados..... | 38 |
| Conclusión..... | 41 |
| Anexos | 43 |

Introducción

En un mundo cada vez más globalizado, la movilidad estudiantil se ha convertido en una experiencia enriquecedora y transformadora para miles de jóvenes europeos. El programa Erasmus, desde su creación en 1987, ha facilitado el intercambio cultural, lingüístico y académico entre estudiantes de diferentes países de la Unión Europea. Sin embargo, junto con las oportunidades que brinda este programa, también surgen una serie de desafíos logísticos y administrativos para los estudiantes que deciden embarcarse en esta aventura educativa en el extranjero.

La presente investigación se centra en el diseño y desarrollo de una aplicación web orientada a proporcionar un servicio integral a los estudiantes Erasmus, con el objetivo de simplificar y agilizar las gestiones administrativas y logísticas que enfrentan durante su estancia en el extranjero, así como poder fomentar la cooperación entre estudiantes. Esta iniciativa no solo pretende facilitar las tareas cotidianas de los estudiantes, sino también fomentar la integración y la solidaridad entre los jóvenes europeos, promoviendo así una mayor cohesión en la comunidad estudiantil del continente.

A través de un enfoque multidisciplinario que combina conceptos de informática, gestión de proyectos y servicios estudiantiles, se abordará el desarrollo de esta aplicación web con el propósito de ofrecer una solución innovadora y eficiente a las necesidades específicas de los estudiantes Erasmus. La aplicación se diseñará teniendo en cuenta aspectos como la accesibilidad, la usabilidad y la seguridad de la información, con el fin de garantizar una experiencia satisfactoria y confiable para sus usuarios.

Mediante este trabajo, se busca contribuir al mejoramiento de la experiencia de los estudiantes Erasmus, facilitando su adaptación y maximizando los beneficios de esta experiencia educativa internacional. Además, se pretende sentar las bases para futuras investigaciones y desarrollos en el campo de las aplicaciones web orientadas al servicio estudiantil, con el potencial de impactar positivamente en la vida de jóvenes europeos en el contexto de la movilidad académica.

Funcionalidades del servicio

El servicio será desarrollado en formato página web, la idea principal de dicho servicio es dar soporte a los estudiantes Erasmus para poder ayudar a estos en todos los trámites burocráticos y logísticos, así como facilitar la cooperación entre dichos estudiantes para que juntos puedan ayudarse a resolver sus potenciales dificultades.

Para ello, el servicio estará diseñado en dos principales bases, la primera será ofrecer al alumnado una información básica para que este pueda consultarla, incluso pudiendo ayudar a este a organizarse con las diferentes tareas que este deberá realizar, la segunda consistirá en crear una red social donde los estudiantes puedan ponerse en contacto con otros estudiantes para poder ayudarse entre ellos, también pudiendo hacer comentarios sobre las diferentes ciudades europeas para dar un feedback de estas.

A continuación, se van a enumerar las funcionalidades del servicio:

1. En primer lugar, el servicio web contará con una funcionalidad dedicada exclusivamente a ayudar al alumnado con los procesos burocráticos que tenga que realizar previamente a su experiencia y durante la propia misma, así bien, podrán acceder a un portal donde tanto su universidad origen como su universidad destino podrán publicar los diferentes procesos abierto que tienen los alumnos, permitiendo a estos visualizar una breve descripción del propio proceso, visualizando también la fecha límite para la realización de este. Además, los usuarios podrán marcar como completado cualquier proceso una vez realizado, de tal manera que esta funcionalidad también le sirva a modo de “checklist” para poder llevar a cabo un control de las tareas ya realizadas y por realizar.
2. La segunda funcionalidad consistirá en poner en contacto a los estudiantes del programa Erasmus. Para ello, una vez se registren indicando la universidad origen, y la universidad destino, se les introducirá directamente en chats con otros estudiantes de su misma universidad origen, chats con estudiantes de su misma universidad destino y chats con estudiantes de su misma ciudad destino. Esto conseguirá que los alumnos cooperen para ayudarse mutuamente, ayudándose con los procesos burocráticos que deban realizar, y también compartiendo información sobre las ciudad destinos de los estudiantes, esto permitirá que estudiantes que se encuentren actualmente en su estancia Erasmus puedan ayudar a estudiantes que vayan a realizar su experiencia en el año posterior, informándoles sobre la mejor manera de encontrar alojamientos, zonas de la ciudad a evitar y aclarando cualquier duda que puedan tener, una vez en la propia estancia Erasmus estos chats podrán poner en contacto a todos los estudiantes en caso de tener cualquier necesidad.

Adicionalmente, esta funcionalidad permitirá contactar de manera individual a otros estudiantes, para ello, contará con una sección que funcionará a modo de chats privados, aquí podrán acceder a los chats que ya tienen con otros usuarios o proceder a buscar usuarios dentro de la página web usando uno de los siguientes filtros, “búsqueda por nombre”, “búsqueda de alumnos que están de Erasmus en una ciudad”,

- “búsqueda de alumnos de una universidad origen concreta”, y “búsqueda de alumnos de una universidad destino/Erasmus concreta”.
3. La siguiente funcionalidad permite a los usuarios el conocer información relevante de las distintas ciudades europeas, con dos potenciales objetivos, el primero ayudar a los futuros estudiantes erasmus a elegir su ciudad destino basándose en el feedback de otros estudiantes, y el segundo facilitar a los estudiantes erasmus información sobre diferentes ciudades europeas con el fin de organizar mejor potenciales viajes que estos realicen. Por ello, esta funcionalidad consistirá en una especie de foro, donde cada ciudad tendrá (al menos de inicio), cuatro subforos, el primero estará dedicado a que los estudiantes posteen comentarios sobre la cultura de ese país, monumentos que ver, museos que visitar..., el segundo estará enfocado en la comida, comida típica de la ciudad, restaurantes donde ir a comer etc, el tercero estará dedicado a los lugares de ocio, donde poder ir a tomar algo, lugares con ambiente etc, y el cuarto estará enfocado en el transporte, cómo es el transporte, que tipos de billetes o bonos hay etc. En cada uno de estos foros los estudiantes podrán comentar lo que deseen para poder ofrecer información a otros estudiantes, además los propios usuarios podrán dar likes a los comentarios que más útiles les parezcan con el fin de que estos aparezcan primero para ayudar a futuros usuarios con la información de mayor utilidad posible.
 4. La última funcionalidad sigue los mismos objetivos que la funcionalidad anterior, sin embargo, en vez de consistir en un foro para realizar comentarios consistirá en un portal de valoraciones, donde los usuarios podrán puntuar diferentes aspectos de una ciudad con el fin de ofrecer información útil a futuros estudiantes.

Método de trabajo

La página web estará programada en fichero contenidos en una misma carpeta y sobre los que se podrá navegar de uno a otro sin mayor dificultad. Estos ficheros estarán programados en varios lenguajes que se especifican a continuación:

- HTML: El lenguaje HTML es un lenguaje de marcado de hipertexto que nos va a permitir programar el frontend de lo que será la página web. Este lenguaje nos permite definir el significado y la estructura que va a tener la página web siendo el contenido más básico de la Web. (MDN web Docs, s.f.)
- CSS: El CSS, cascading style sheet, permite crear páginas web más atractivas mejorando el diseño de esta, permitiendo cambiar los tamaños, los colores, las fuentes etc, de contenidos concretos dentro de la página web.
- PHP: Este lenguaje también conocido como Hypertext Preprocessor es un lenguaje de código abierto que tiene la peculiaridad de poder ser incrustado de manera sencilla en HTML. Este lenguaje nos va a permitir programar todo el backend del servicio para poder cumplimentar las funcionalidades de la aplicación. (PHP.net, s.f.)
- SQL: Para la gestión de la base de datos necesaria para poder llevar a cabo el desarrollo del servicio web se va a contar con un base de datos SQL, esta es una base de datos relacional que sigue por lo tanto un formato de tablas, siendo probablemente el lenguaje de base de datos relacionales más popular actualmente.

Diseño del servicio web

En el siguiente apartado se va a realizar una explicación extensa del procedimiento de implementación del servicio web, en el que se explican las funcionalidades de los diferentes archivos de códigos y la interacción de estos dentro de la aplicación, así como el diseño de la base de datos que da sustento a la aplicación y como interactúa el servicio con esta misma.

Diseño de la Base de Datos

Para la implementación del servicio web se ha requerido la creación de una base de datos que sigue un modelo relacional, para ello, se ha utilizado MySQL para la implementación de esta.

El modelo relacional es un modelo basado en tablas, en el que se definen entidades, atributos de las entidades y la relación que va a existir entre estas. Para poder vislumbrar esto de modo visual se ha realizado un diseño gráfico:

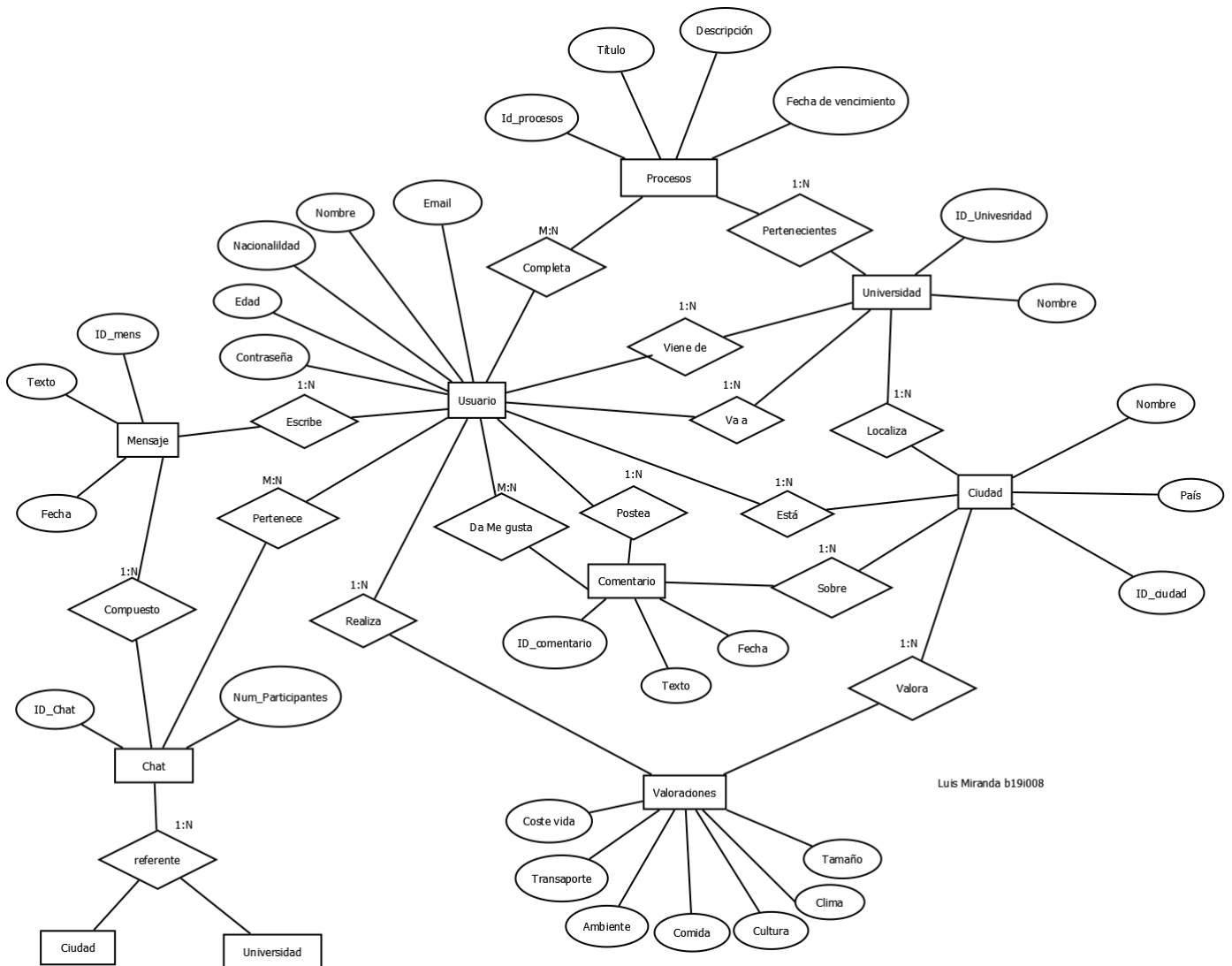


Ilustración 1: Diseño BBDD

Entidades

En primer lugar, vamos a definir las entidades que forman parte del diseño ilustrado en la ilustración 1.

- **Usuario:** La entidad usuario va a permitir almacenar los detalles de los usuarios que se hayan registrado en el servicio web, los atributos de esta entidad recopilan la información necesaria a recopilar de los usuarios:
 1. Email: El email con el que se registrarán los usuarios, este atributo es obligatorio y además hará la función de clave primaria, será de tipo varchar.
 2. Nombre: En este atributo se almacenará el nombre del usuario, no puede ser null y será de tipo varchar.
 3. Passw: En este atributo se almacenará la contraseña que el usuario establezca para su cuenta, no puede ser null y será tipo varchar.

- **Ciudad:** La entidad ciudad va a permitir almacenar los detalles de las ciudades europeas donde se realice el programa erasmus, los atributos de esta entidad recopilan la información necesaria acerca de las ciudades:
 1. Id_ciudad: El id_ciudad será un atributo que nos permita asignar un id identificativo a cada ciudad, por ello hará de clave primaria y no podrá ser null, cabe destacar que será tipo auto_increment, lo que permite no tener que indicar un id concreto al hacer la inserción de los datos.
 2. Nombre: En este atributo se almacenará el nombre de la ciudad, no puede ser null y será tipo varchar.
 3. País: En este atributo se almacenará el país en el que esté dicha ciudad, no puede ser null y será tipo varchar.

- **Universidad:** La entidad universidad va a permitir almacenar las universidades europeas que participen activamente en el programa erasmus, los atributos de esta entidad recopilan la información necesaria acerca de las universidades:
 1. Id_universidad: El id_universidad será un atributo que nos permita asignar un id identificativo a cada universidad, por ello hará de clave primaria y no podrá ser null, cabe destacar que será tipo auto_increment, lo que permite no tener que indicar un id concreto al hacer la inserción de los datos.
 2. Nombre: En este atributo se almacenará el nombre de la universidad, no puede ser null y será tipo varchar.

- **Comentario:** La entidad comentario va a permitir el almacenamiento de los comentarios que los usuarios realicen sobre las diferentes ciudades, los atributos de esta entidad recopilan la información necesaria acerca de los comentarios:

1. **Id_comentario:** El id_comentario será un atributo que nos permita asignar un id identificativo a cada comentario, por ello hará de clave primaria y no podrá ser null, cabe destacar que será tipo auto_increment, lo que permite no tener que indicar un id concreto al hacer la inserción de los datos.
 2. **Texto:** Este atributo nos va a permitir almacenar el texto del comentario que el usuario realice, será del tipo varchar.
 3. **Fecha:** Este atributo nos va a permitir almacenar la fecha y la hora a la que se ha realizado el comentario, será de tipo timestamp y tendrá un valor por defecto que será Current_Timestamp lo que permitirá que este valor se genere solo y no haya que indicarlo a la hora de insertar los datos.
 4. **Tipo:** Este atributo nos permitirá diferenciar los comentarios según su tipo, existen cuatro tipos definidos en el servicio, “Comida”, “Cultura”, “Fiesta” y “Transporte”, por ello para este atributo se realizará un “Check” para comprobar que el valor de este coincida con alguno de esos cuatro, este atributo no podrá ser null.
- **Chats:** La entidad chats va a permitir almacenar la información necesaria relativa a los chats, los atributos de esta entidad recopilan la información necesaria acerca de los chats:
1. **Id_Chat:** El id_chat será un atributo que nos permita asignar un id identificativo a cada chat, por ello hará de clave primaria y no podrá ser null, cabe destacar que será tipo auto_increment, lo que permite no tener que indicar un id concreto al hacer la inserción de los datos.
 2. **Num_participantes:** Este atributo nos permitirá conocer el número de participantes que hay en un chat.
- **Mensaje:** La entidad mensaje permite almacenar la información necesaria sobre los mensajes que se realicen en los chats, los atributos de esta entidad recopilan la información necesaria acerca de los mensajes:
1. **Id_mensaje:** El id_mensaje será un atributo que nos permita asignar un id identificativo a cada mensaje, por ello hará de clave primaria y no podrá ser null, cabe destacar que será tipo auto_increment, lo que permite no tener que indicar un id concreto al hacer la inserción de los datos.
 2. **Texto:** Este atributo nos permitirá almacenar el texto del mensaje, será de tipo varchar.
 3. **Fecha:** Este atributo nos va a permitir almacenar la fecha y la hora a la que se ha escrito el mensaje, será de tipo timestamp y tendrá un valor por defecto que será Current_Timestamp lo que permitirá que este valor se genere solo y no haya que indicarlo a la hora de insertar los datos.

- **Valoraciones:** Esta entidad va a permitir almacenar la información necesaria acerca de las valoraciones que realicen los usuarios de las distintas ciudades europeas, cabe destacar que para esta entidad la clave primaria no será un atributo propio sino la combinación de dos atributos foráneos, que indican el id de la ciudad que se está valorando y el email del usuario que realiza la valoración, los atributos de esta entidad recopilan la información necesaria acerca de las valoraciones:
 1. Coste_vida: Este atributo recopilará el valor que el usuario le haya dado al coste de vida, será de tipo int.
 2. Transporte: Este atributo recopilará el valor que el usuario le haya dado al transporte, será de tipo int.
 3. Ambiente: Este atributo recopilará el valor que el usuario le haya dado al ambiente, será de tipo int.
 4. Comida: Este atributo recopilará el valor que el usuario le haya dado a la comida, será de tipo int.
 5. Cultura: Este atributo recopilará el valor que el usuario le haya dado a la cultura, será de tipo int.
 6. Clima: Este atributo recopilará el valor que el usuario le haya dado al clima, será de tipo int.
 7. Tamaño: Este atributo recopilará el valor que el usuario le haya dado al tamaño, será de tipo int.

- **Procesos:** En esta entidad se va a poder almacenar la información necesaria acerca de los procesos que debe realizar el usuario, los atributos de esta entidad recopilan la información necesaria acerca de los procesos:
 1. Id_procesos: El id_procesos será un atributo que nos permita asignar un id identificativo a cada proceso, por ello hará de clave primaria y no podrá ser null, cabe destacar que será tipo auto_increment, lo que permite no tener que indicar un id concreto al hacer la inserción de los datos.
 2. Título: Este atributo nos permitirá almacenar el título que recibe el proceso en cuestión, no podrá ser null y será tipo varchar.
 3. Descripción: Este atributo permite añadir información necesaria acerca del proceso en cuestión, será de tipo varchar.
 4. Fecha_vencimiento: Este atributo permite almacenar la fecha límite para realizar este proceso, será de tipo date.
 5. Tipo: Es un atributo que nos permite identificar si el proceso concreto es para los alumnos cuyo origen erasmus sea la universidad que publica el proceso o si es para alumnos cuyo destino erasmus sea dicha universidad, este atributo puede tener dos valores “origen” o “destino” y para ello se realizará un check sobre el valor de este, no podrá ser null.

Relaciones que generan tabla

En esta sección se van a definir las relaciones entre entidades que generan una tabla propia en la base de datos, esto se debe a que son relaciones tipo “M:N”.

- **Da me gusta:** La siguiente relación es establecida entre la entidad comentario y la entidad usuario, en ella se recogen los likes que dan los usuarios a los comentarios que más útiles les han parecido, es una relación M:N debido a que un usuario puede dar likes a más de un comentario y un comentario puede recibir likes de más de un usuario, la tabla generada contará con dos claves foráneas, email_usuario e id_comentario, que en su conjunto actuaran como clave primaria.
- **Pertenece:** La siguiente relación es establecida entre la entidad usuario y la entidad chat, en ella se recogen los usuarios que participan activamente en ciertos chats, es una relación M:N ya que un usuario puede participar en más de un chat, y en un chat puede haber más de un usuario, la tabla generada contendrá dos claves foráneas, id_chat y email_usuario, que en su conjunto actuaran como clave primaria.
- **Referente:** La siguiente relación es una relación “triple” que se establece entre chat, ciudad y universidad, no obstante, no es del todo cierto esto ya que es una relación que permite entender si un chat es referente a un chat de usuarios de una ciudad concreta o a un chat de usuarios de una universidad concreta y en caso de ser de esta segunda saber si es un chat referente a los usuarios que tienen como origen la universidad dicha o si la tienen como destino. Por ello, id_chat será obligatorio y no podrá ser null ya que realizará la función de clave primaria, sin embargo, el id_referente_ciudad o el id_referente_universidad podrán ser null, no obstante, al menos uno deberá tener valor, además, se indicará a través del atributo “Id_tipo_refencia” si se está referenciando una ciudad o una universidad comprobando a través de un check que el valor otorgado sea o “ciudad” o “universidad”, en caso de que sea un chat referente a una universidad el atributo de la relación denominado “tipo_universidad” tendrá que tener un valor entre origen o destino, para ellos se realizará otro check que compruebe que el valor establecido para ese atributo corresponde a uno de ellos, no obstante, este valor también puede ser null en caso de que se referencia una ciudad.
- **Completa:** La siguiente relación es establecida entre la entidad usuario y la entidad proceso, en ella se recogen los usuarios que han ido completando los procesos que han de realizar, es una relación tipo M:N ya que un usuario puede completar más de un proceso y un proceso puede ser completado por más de un usuario, la tabla generada contendrá dos claves foráneas, id_proceso y email_usuario, que en su conjunto actuaran como clave primaria.

Relaciones que no generan tabla.

A continuación, se van a definir las relaciones que se dan entre entidades y que no generan tabla al ser relaciones 1:N:

- **Localiza:** Esta es una relación establecida entre la entidad ciudad y la entidad universidad en ella se recoge en que ciudad está localizada la universidad, es una relación 1:N ya que una universidad solo puede estar

localizada en una ciudad, por ello, se genera una clave foránea, *id_ciudad*, en la tabla universidad.

- **Viene de:** Esta relación se establece entre la entidad usuario y la entidad universidad y sirve para indicar la universidad origen de un usuario, es una relación 1:N ya que un usuario solo puede tener una universidad origen, por ello, se genera una clave foránea, *id_universidad_origen*, en la tabla usuario.
- **Va a:** Esta relación es idéntica a la anterior, pero hace referencia a la universidad destino del usuario, la clave foránea que se genera en la tabla usuario se denomina *id_universidad_destino*.
- **Está:** Esta relación se establece entre la entidad usuario y la entidad ciudad y sirve para identificar la ciudad a la que va a ir el usuario en su experiencia erasmus, es una relación 1:N ya que el usuario solo puede tener una ciudad destino, por ello, se genera una clave foránea *id_ciudad* en la tabla usuario.
- **Postea:** Esta relación se establece entre la entidad usuario y la entidad comentario y sirve para identificar qué usuario ha realizado un comentario, es una relación 1:N ya que un comentario solo puede ser escrito por un usuario concreto, por ello, se genera una clave foránea *email_usuario* en la tabla comentario.
- **Sobre:** Esta relación se establece entre la entidad comentario y la entidad ciudad y sirve para identificar a que ciudad se está refiriendo el comentario en cuestión, es una relación 1:N ya que un comentario se refiere a una ciudad concreta, por ello, se genera una clave foránea *id_ciudad* en la tabla comentario.
- **Compuesto:** Esta relación se establece entre la entidad chat y la entidad mensaje y sirve para identificar a qué chat pertenece un mensaje concreto, es una relación 1:N ya que un mensaje solo puede pertenecer a un chat concreto, por ello, se genera una clave foránea *id_chat* en la tabla mensajes.
- **Escribe:** Esta relación se establece entre la entidad mensaje y la entidad usuario y sirve para identificar que usuario a escrito un mensaje concreto, es una relación 1:N debido a que un mensaje solo puede ser escrito por un usuario, por ello, se genera una clave foránea *email_usuario* en la tabla mensajes.
- **Realiza:** Esta relación se establece entre la entidad valoraciones y la entidad usuario y sirve para identificar que usuario ha realizado la valoración, es una relación 1:N ya que una valoración solo puede ser realizada por un usuario concreto, por ello, se genera una clave foránea *email_usuario* en la tabla valoraciones.

- **Valora:** Esta relación se establece entre la entidad valoraciones y la entidad ciudad y sirve para identificar a que ciudad hace referencia la valoración, es una relación 1:N ya que una valoración solo puede hacer referencia a una ciudad, por ello, se genera una clave foránea *id_ciudad* en la tabla valoraciones.
- **Pertenecientes:** Esta relación se realiza entre la entidad procesos y la entidad universidad y sirve para conocer la universidad que ha publicado dicho proceso, es una relación 1:N ya que un proceso solo puede ser publicado por una universidad concreta, por ello, se genera una clave foránea *id_universidad* en la tabla procesos.

Diseño de la página web

Como se ha mencionado previamente, para el diseño de la página web se va a hacer uso de tres lenguajes, para el backend se ha usado el lenguaje “php”, y para el diseño de la página web el lenguaje “HTML”, para la decoración de esta se ha utilizado “CSS”, con el fin de obtener una interfaz atractiva para los usuarios. A pesar de que se utilicen estos tres lenguajes todos los archivos que componen la página web son del tipo .php, dentro de estos se incrustan los códigos correspondientes en HTTP y CSS.

A continuación, se va a realizar una descripción de los archivos y su funcionalidad dentro del sistema:

PantallaInicio.php

Este script es el primer script en ejecutarse cuando se accede al servicio web, y ofrecerá la página de inicio al usuario, su contenido es sencillo, cuenta con un header en el cual se ofrecerá un título para la página web, y botones de acceso a las diferentes funcionalidades que son ofrecidas en la página web y que quedaron resumidas al comienzo de la memoria, contará también con un botón para que el usuario pueda acceder al inicio de sesión o en caso de que el usuario tenga la sesión iniciada el botón ofrecerá la posibilidad de cerrar la sesión.

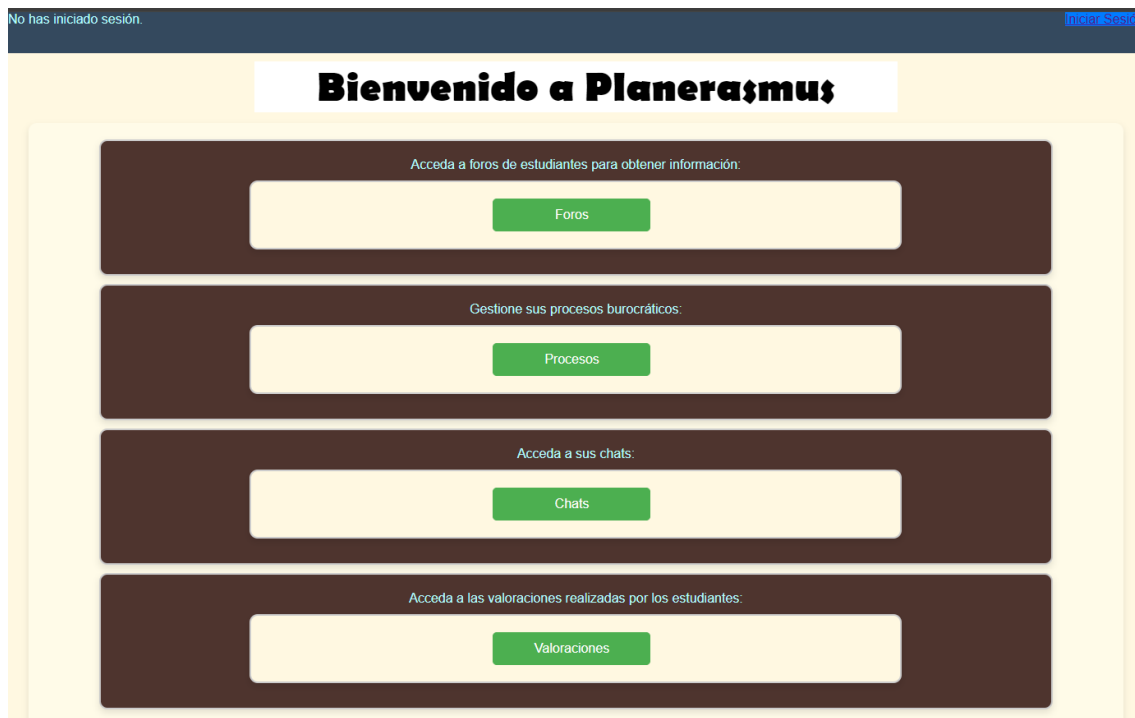


Ilustración 2: Pantalla de Inicio

PantallaInicioSesion.php

Este script es accedido desde cualquier punto de la página web a través de un botón localizado en la esquina superior derecha con el texto “iniciar sesión”. Como su nombre indica desde este script el usuario tiene la capacidad de iniciar sesión en su cuenta o de acceder al proceso de registro en la página web.

Este script está diseñado a modo de formulario, en la cabecera contendrá un título que indica que estamos antes un script de inicio de sesión y a continuación se le ofrecerá al usuario un campo email y un campo contraseña que el usuario podrá rellenar con sus datos de inicio de sesión, el formulario concluirá con un botón que tenga la función de submit realizando un método de tipo post sobre el script “*inicio_sesion.php*” con los valores registrados en los campos indicados, cabe destacar que ambos campos son *required*, es decir, el botón para iniciar sesión no tendrá ningún efecto si no están ambos campos rellenos.

Por otro lado, contendrá un botón con el texto “Registrarse” que permitirá al usuario acceder al script de registro.

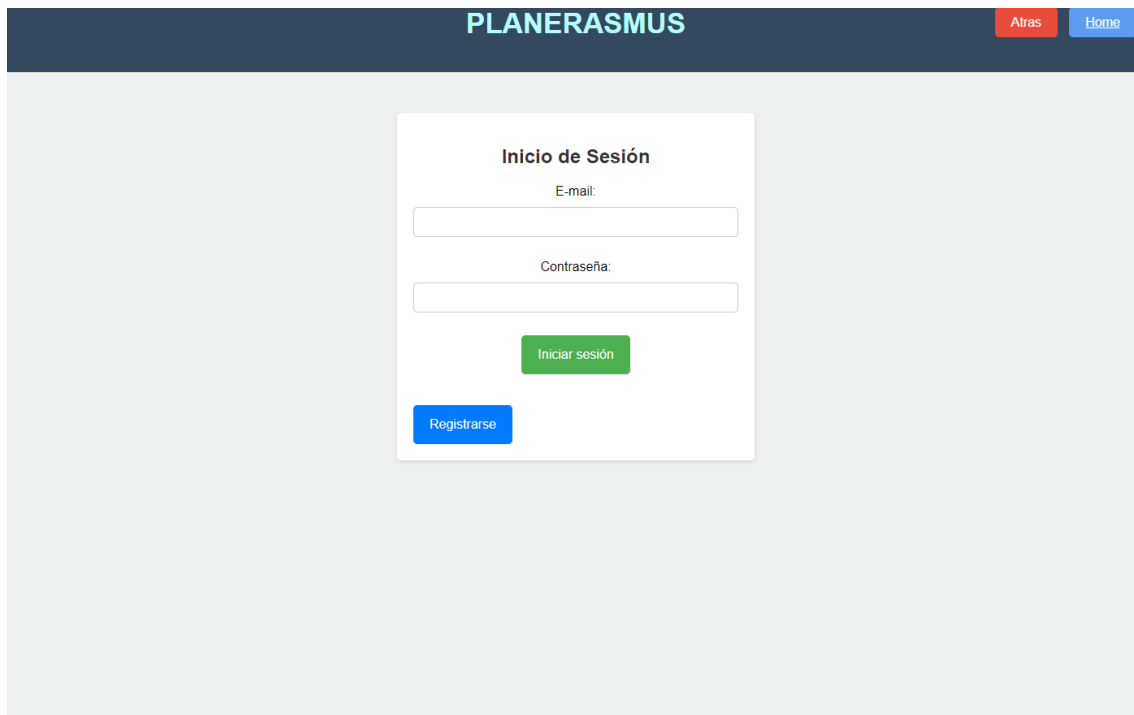


Ilustración 3: Pantalla de Inicio de Sesión

Inicio_sesion.php

Este script tendrá funcionalidad plenamente de backend ya que no contendrá ninguna inserción de código HTML. La funcionalidad de este consistirá en verificar los datos introducidos por el usuario e iniciar sesión a este si procede.

Como se le ha llamado a través de un método post lo primero que hace el script es recibir el email del usuario y comprobar que no sea vacío, esto es una comprobación para una seguridad extra, en principio no se debería poder acceder a este script sin completar este campo por lo mencionado en la descripción del script anterior.

Una vez comprobado que no es vacío lo que se realiza es una consulta a la base de datos en la que se va a solicitar la contraseña del usuario que tenga dicho email.

Llegados a este punto pueden pasar dos cosas o bien se reciba la contraseña, en ese caso lo que se procederá es a compararla con la facilitada por el usuario, en caso de que coincidan se iniciará sesión, en caso contrario se volverá al script anterior indicando que la contraseña es incorrecta, en el caso de que no se obtenga la contraseña se volverá al script anterior indicando que no existe registro de ningún usuario con dicho email.

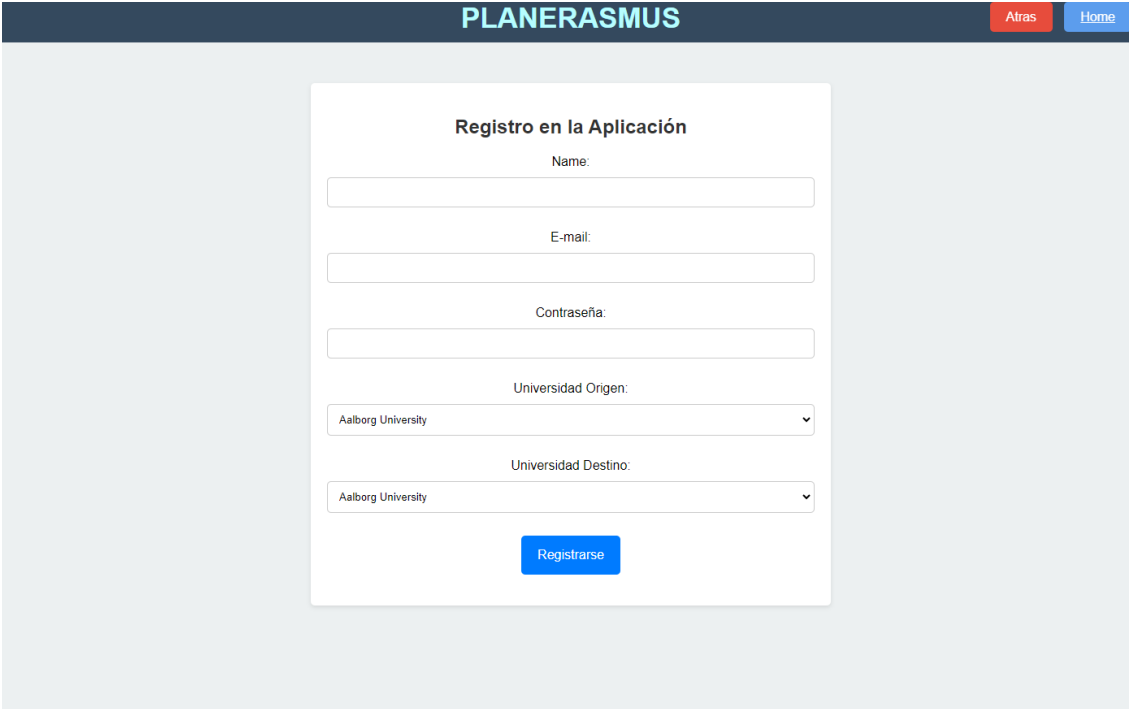
Cerrarsesion.php

El siguiente script es muy sencillo, funcionará a modo de backend y lo único que hará será eliminar el registro de la sesión iniciada actualmente, se puede acceder a él desde cualquier script siempre y cuando haya una sesión iniciada.

Registro_usuario.php

Este script tendrá la funcionalidad de ofrecer al usuario un formulario de registro y por consiguiente, se basará en un formulario que ejecutará un método post sobre el script “*input_datos.php*”.

En este formulario se le solicitará al usuario que indique los siguientes campos: Nombre, Email, Contraseña, Universidad Origen y Universidad Destino, para las universidades se realizará una consulta en la base de datos recuperando todas las universidades que existen en esta para mostrárselas al usuario a modo de lista con el fin de que haga su selección a través de dicha lista.



The screenshot shows a web application interface for registration. At the top, there is a dark blue header with the text "PLANERASMUS" in white. To the right of the header are two buttons: "Atras" (red) and "Home" (blue). Below the header is a light gray background containing a white registration form. The form is titled "Registro en la Aplicación" and has the following fields: "Name:" with a text input field; "E-mail:" with a text input field; "Contraseña:" with a text input field; "Universidad Origen:" with a dropdown menu showing "Aalborg University"; and "Universidad Destino:" with a dropdown menu also showing "Aalborg University". At the bottom of the form is a blue button labeled "Registrarse".

Ilustración 4: Pantalla de Registro

Input_datos.php

El siguiente script tiene una funcionalidad plenamente de backend, su objetivo será proceder al registro del usuario en la página web.

Para la inserción de los usuarios en la base de datos el primer paso será recuperar los valores de los campos indicados en el formulario del script anterior y recibidos a través de un método post, una vez recibido estos se harán una comprobación antes de introducir al usuario en la base de datos, para todos los datos se comprobarán que no sean vacíos y para el email además se comprobará a través de una consulta en la base de datos que no exista ningún usuario registrado previamente en la página web con dicho email, en caso de que algún dato no sea correcto se volverá al script anterior sin introducir a ningún usuario y devolviendo el error correspondiente.

Una vez comprobado que los datos son correctos, lo primero que se hará será obtener el id de la ciudad destino a través de la universidad destino del usuario, tras esto se realizará una inserción de los datos en la tabla usuario, con todos los datos recogidos previamente.

Finalmente el usuario quedará registrado en la BBDD, no obstante, adicionalmente se introduce al usuario como miembro de los chats correspondientes a su universidad origen, su universidad destino y su ciudad, para ello se obtendrá el id de los chats a través de una consulta a la tabla “referente” que asocia los chats con las universidades y las ciudades, y tras esto se añade un registro en la tabla “pertenece” con el email del usuario y el id del chat, además se actualiza el número de participantes del chat sumándole uno.

Acceso_procesos.php

Desde la página de inicio se tendrá acceso a este Script a través del botón “Procesos en curso”, este script servirá como script para la bienvenida de los usuarios a la funcionalidad donde se les permitirá gestionar sus procesos. Esta interfaz contará con dos botones, el primero permitirá acceder al usuario a los procesos que ha de realizar marcados por su universidad origen, y el segundo le permitirá acceder a los procesos que ha de realizar marcados por su universidad destino.



Ilustración 5: Pantalla de acceso a Procesos

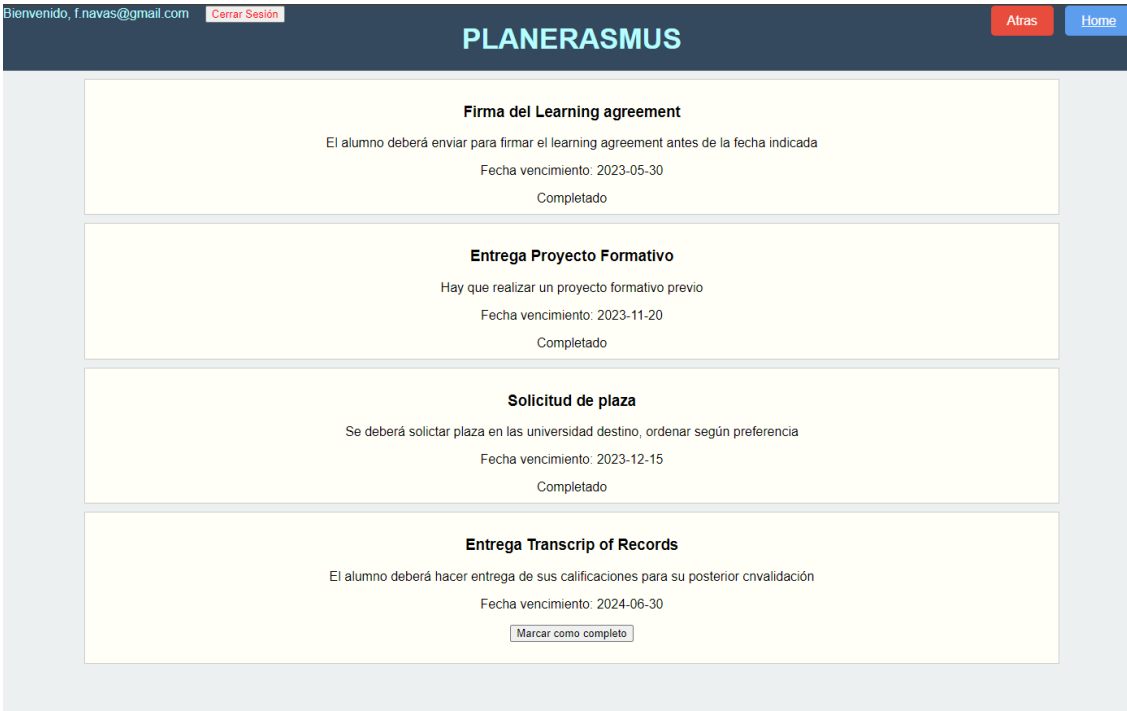
DevolverProcesos.php

Este script será accedido desde el script descrito previamente y permitirá a los usuarios visualizar los procesos que han de realizar, cabe destacar que la llamada a este script por parte del script anterior viene acompañada de un parámetro que se denominará “tipo” y que nos permitirá conocer si el usuario desea acceder a los procesos de su universidad origen o si por el contrario los usuarios desean acceder a los de su universidad destino.

El primer paso que se va a realizar en este script es la comprobación de si el usuario ha iniciado sesión o no, ya que para poder mostrarle sus procesos es necesario conocer de que usuario se trata, en caso de no haber iniciado sesión se le redirigirá automáticamente al script de inicio de sesión.

Una vez exista una sesión activa se va a proceder a realizar una conexión con la base de datos, tras establecer la conexión con esta se realizará una consulta en la que se realizará un inner join entre tres tablas: “procesos”, “universidad” y “usuario”, se unirá la tabla procesos y la tabla universidad a través de la clave primaria de universidad y la clave foránea que tienen los procesos en referencia a la universidad que ha publicado dicho proceso, por otro lado se hará un inner join entre la clave primaria de la universidad y la clave foránea de la tabla usuario que hace referencia a la universidad origen o destino del usuario, según el parámetro indicado en la llamada, filtrando los resultados por el tipo del proceso, y por el email del usuario, y ordenando los resultados por la fecha de vencimiento.

Una vez realizada la consulta se imprimen los datos de los procesos en bloque, cada bloque contendrá el título, la descripción, la fecha de vencimiento y se realizará una consulta sobre la tabla “completa” donde se consultará si existe un registro del proceso y el usuario, es decir, si el usuario ya ha marcado el proceso como completado, en caso de que no lo haya hecho, se imprimirá un botón para que usuario pueda marcar como completado, llamando al script “*marcar_proceso.php*” o si ya ha sido marcado se imprimirá el texto “Completado”.



The screenshot shows the PLANERASMUS web application interface. At the top, there is a navigation bar with the user's name and email (Bienvenido, f.navas@gmail.com), a 'Cerrar Sesión' button, and the application name 'PLANERASMUS'. There are also 'Atras' and 'Home' buttons. The main content area displays four process blocks, each with a title, description, due date, and completion status:

- Firma del Learning agreement**: El alumno deberá enviar para firmar el learning agreement antes de la fecha indicada. Fecha vencimiento: 2023-05-30. Completado.
- Entrega Proyecto Formativo**: Hay que realizar un proyecto formativo previo. Fecha vencimiento: 2023-11-20. Completado.
- Solicitud de plaza**: Se deberá solicitar plaza en las universidad destino, ordenar según preferencia. Fecha vencimiento: 2023-12-15. Completado.
- Entrega Transcript of Records**: El alumno deberá hacer entrega de sus calificaciones para su posterior cnvalidación. Fecha vencimiento: 2024-06-30. Includes a 'Marcar como completo' button.

Ilustración 6: Pantalla visualización de procesos

Marcar_proceso.php

Este script no contiene inserción de código en HTML con lo que su funcionalidad es a modo de backend en su totalidad. Este script permite al usuario marcar un proceso como completado, es llamado a través del script anterior “*devolverProcesos.php*”, recibe una consulta de tipo POST, que fue creada en el script anterior, recoge el valor dado a la variable “proceso” y lo que realiza es un insert en la tabla “completa” en la que introduce una nueva entrada que contiene el id del proceso recogido en la variable mencionada y el email del

usuario que tiene la sesión iniciada, una vez finalizado redirige automáticamente al script anterior.

Foros.php

El siguiente script permite al usuario acceder a los foros sobre una ciudad determinada, este script es accedido directamente desde la pantalla de inicio.

Este script mostrará al usuario un seleccionable para que pueda seleccionar la ciudad sobre la cual quiera hacer la consulta, para ello, se declarará un formulario con un campo select que mostrará las opciones seleccionables gracias a la realización de una consulta sobre la base de datos previa en la cual se realizará un select sobre la tabla “ciudad” y se imprimirán los resultados obtenidos.

Una vez seleccionado este script tendrá un botón con el texto “Acceso” que permitirá llamar al script “*acceso_foros.php*” con un método post.



Ilustración 7: Pantalla de selección de la ciudad sobre la que consultar los foros

Acceso_foros.php

Este script es bastante sencillo, comprobará la ciudad que se le ha pasado desde el script anterior a través de un método post, la almacenará y mostrará al usuario cuatro botones para que puedan elegir el foro al que quieran acceder, actualmente se cuenta con cuatro foros diferenciados (uno por cada botón), tendremos un foro de comida, otro de cultura, otro de fiesta y otro de transporte. Todos los botones redirigirán hacia el script “DevolverComentarioF.php”, introduciendo un parámetro a la llamada denominado “tipo” que recibirá el valor dependiendo de a qué foro quieran acceder los usuarios.

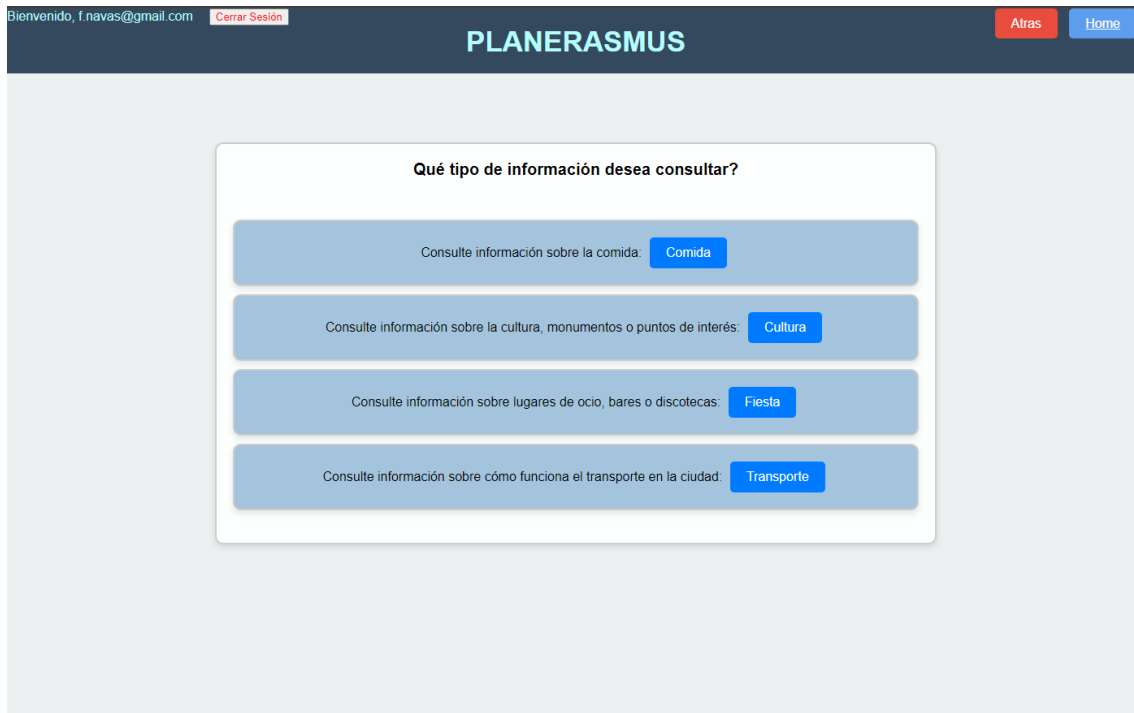


Ilustración 8: Pantalla para seleccionar el topic del foro a ver

DevolverComentarioF.php

El siguiente script va a permitir al usuario visualizar los comentarios realizados en un foro determinado por parte de otros usuarios, para ello, va a ser imprescindible que el usuario haya iniciado sesión en la página web ya que podrá realizar ciertas acciones detalladas a continuación.

Lo primero que se realizará en el script es la recuperación del parámetro tipo pasado desde el script previo, para la recuperación de estos parámetros se realiza un método get, también vamos a pasar a recuperar el valor almacenado de la ciudad sobre la cual se está realizando la consulta.

Una vez realizado este se procederá a realizar una consulta en la base de datos para recuperar todos los comentarios posteados por parte de los usuarios en dicho foro, cabe destacar que los comentarios vendrán acompañados por los likes que han recibido estos, estos likes son almacenados en otra tabla "likes" que asociada los comentarios con los usuarios que han dado like a dichos comentarios, por ello, a la hora de realizar la consulta se realizará un inner join entre las tablas comentario y likes, igualando la clave primaria de los comentarios con la clave foránea de los likes denominada "id_comentario", se filtrará por el id de la ciudad y por el tipo de comentarios que ha requerido el usuario, se agruparan por el id de los comentarios de la tabla comentario con el fin de que no aparezcan repeticiones de comentarios debido al inner join y se realizará un count de estos con el fin de determinar el número de likes de estos, finalmente los comentarios serán ordenados por número de likes en primera instancia y por fecha de publicación es segunda, además, el número de comentarios estará limitado a través de una variable almacenada denominada "num_coments" cuyo valor inicial será de diez.

Una vez recuperados los comentarios se imprimirán dichos comentarios indicando la fecha de publicación, el usuario que lo publicó, el número de likes y el comentario en sí.

Se añadirá una funcionalidad que consistirá en dar likes al comentario, para ello se realizará un formulario que lanzará un script “*dar_like.php*” con un método post. En caso de que el usuario no haya iniciado sesión o sea el mismo que escribió el comentario, el botón de likes estará escondido, en caso contrario se realizará una consulta sobre la tabla “likes” filtrada con el usuario y el id del comentario con el fin de comprobar si el usuario ya ha dado like al comentario, en caso de que ya lo haya hecho el botón recibirá el texto de “Quitar me gusta” si no le ha dado like recibirá el texto de “Me gusta”, para cualquiera de los dos caso el botón podrá ser seleccionado por el usuario lanzando el script mencionado previamente.

Por último, se contará con dos botones adicionales, uno con el texto “Añada un comentario” que lanzará un script “*añadir_comentario.php*” con un método post que pasará el tipo de comentario que debe ser, y el otro con el texto “Mostrar más comentario” que servirá para cambiar el valor de la variable “num_comentario” sumándola diez y relanzará el script.

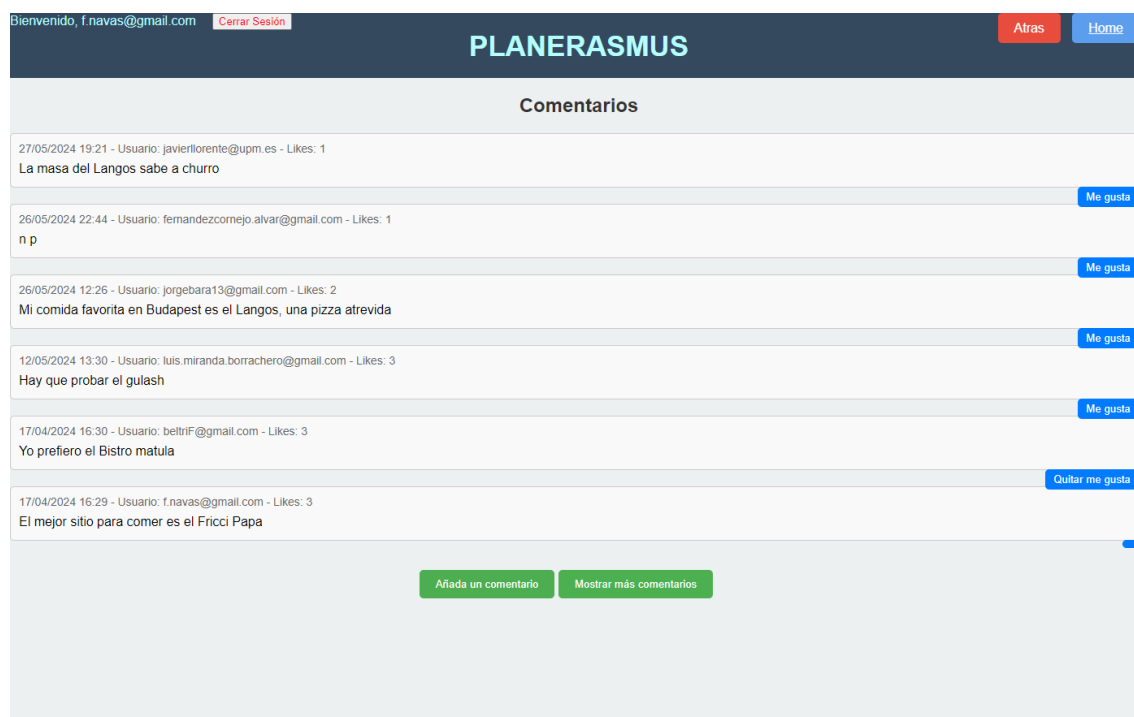


Ilustración 9: Pantalla de comentarios de los foros

Dar_like.php

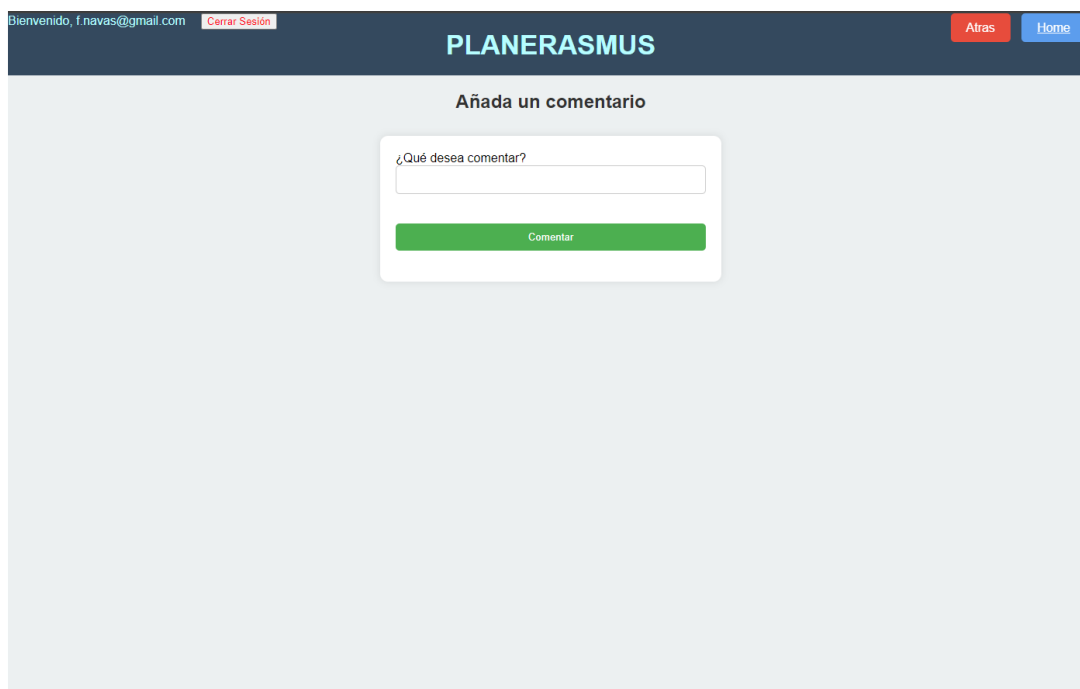
El siguiente script tendrá una funcionalidad exclusiva de backend, su funcionalidad consistirá en poder dar me gusta o quitar los me gusta a los comentarios que realizan los usuarios en el foro.

Para hacer esto, lo primero que se realizará será la recuperación del parámetro “id_comentario” pasada gracias al método post realizado en el script “Devolver_comentarioF.php”, a continuación, se realizará una consulta sobre la tabla likes filtrando por el comentario y el email del usuario que tiene iniciada

la sesión con el fin de comprobar si ya existe un registro de esta combinación, es decir, si el usuario había dado previamente like y ahora quiere eliminarlo. En caso de que ya existiera el registro se eliminaría de la tabla, y en caso de que no se insertaría en la tabla.

Añadir_comentario.php

El siguiente script tiene como funcionalidad ofrecer al usuario una interfaz donde pueda realizar comentarios sobre el foro en el que se encuentra, para ello contará tan solo con un formulario que generará un método post y que llamará al script “añadir_comentarioBE.php”. El formulario estará compuesto de un texto, un cuadro donde los usuarios podrán escribir lo que deseen y un botón para postear el comentario.



The screenshot shows a web application interface. At the top, there is a dark blue header with the text 'PLANERASMUS' in white. On the left, it says 'Bienvenido, f.navas@gmail.com' and 'Cerrar Sesión'. On the right, there are two buttons: 'Atras' (red) and 'Home' (blue). Below the header, the main content area has a light gray background. The heading 'Añada un comentario' is centered. Below this heading is a white form box containing a text input field with the placeholder text '¿Qué desea comentar?' and a green button labeled 'Comentar'.

Ilustración 10: Pantalla para añadir comentarios en el foro

Añadir_comentarioBE.php

El siguiente script será un script que funcione a modo de backend, su funcionalidad consistirá en la introducción de un comentario dentro de la base de datos.

Lo primero que hará el script será recuperar el valor del parámetro “texto” que fue enviado a través de un método post desde el script “añadir_comentario.php”, acto seguido procederá a insertar el comentario en la tabla “comentario”.

Valoraciones.php

El siguiente script servirá de acceso a la funcionalidad descrita al principio de la memoria que consistirá en una sección donde se podrán valorar diferentes aspectos de las ciudades, así como obtener las valoraciones medias que ha realizado la gente.

Este script es accedido desde la página principal y va a mostrar al usuario un formulario donde el usuario podrá seleccionar la ciudad sobre la cual quiere

obtener las valoraciones, este formulario llamará al script “*devolverValoracionesF.php*” realizando un método post.

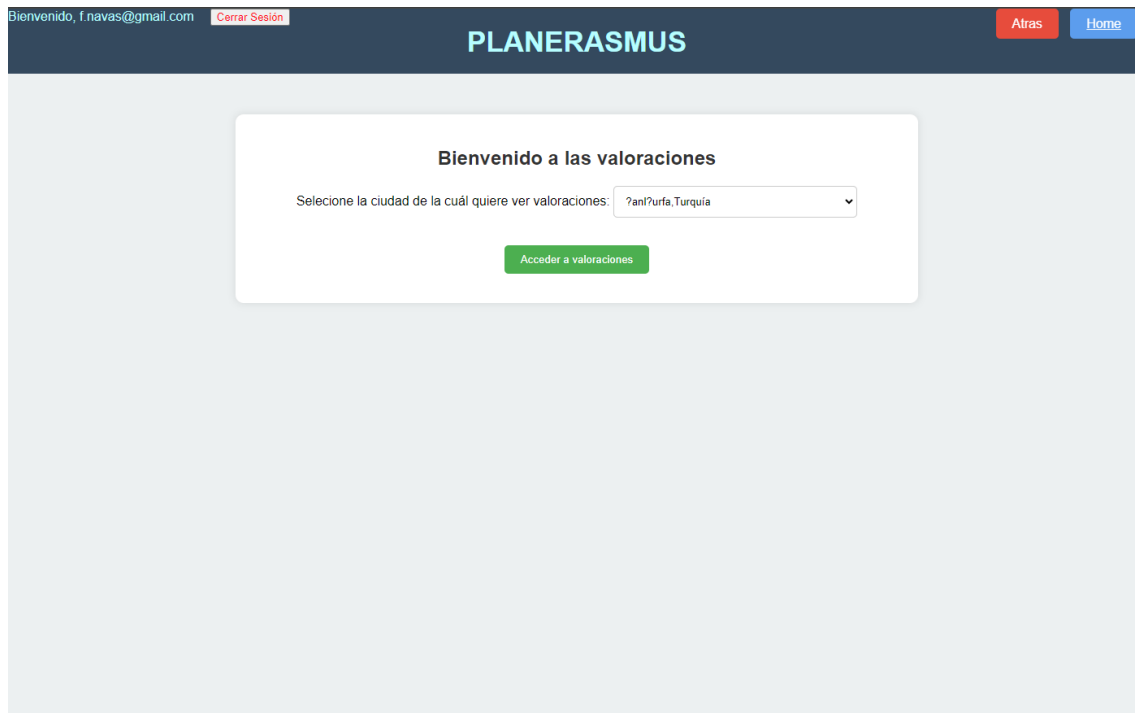


Ilustración 11: Pantalla para la selección de la ciudad sobre la que consultar las valoraciones

DevolverValoracionesF.php

Tras la selección de la ciudad sobre la cual se quieren consultar las valoraciones se accederá al siguiente script donde el usuario podrá visualizar las notas medias que recibe la ciudad en ciertos aspectos que son: el coste de vida, el transporte, el ambiente, la comida, la cultura, el clima y el tamaño de esta.

Para consultar esto lo que se realiza en primer lugar es la recuperación del parámetro lanzado por el método post del script anterior denominado “Ciudad” y que servirá para obtener el id de la ciudad sobre la cual se está realizando la consulta. Una vez obtenido dicho id se procede a realizar una consulta en la base de datos sobre la tabla “valoraciones” filtrando por el id de la ciudad y obteniendo la media de los atributos mencionados previamente, para proceder a imprimirlos por la pantalla para la visualización por parte del usuario.

Se incluirá también un botón que realizará una llamada al script “*añadir_valoraciones.php*” con un método post.



Ilustración 12: Pantalla de visualización de valoraciones

Añadir_valoraciones.php

EL siguiente script permitirá al usuario realizar las valoraciones sobre la ciudad que consideren, este script consistirá principalmente en un formulario donde los usuarios se van a poder seleccionar una nota de 1 al 10 para cada uno de los atributos mencionados previamente. No será obligatorio el dar nota a todos los aspectos presentados.

Una vez seleccionadas las notas se realizará una llamada al script “añadir_valoraciones.php” utilizando en método post para ello.

Bienvenido, f.navas@gmail.com Cerrar Sesión

PLANERASMUS

Atrás Home

Añada sus valoraciones

Cómo valora el coste de vida?:
0

Cómo valora el transporte de la ciudad?:
0

Cómo valora el ambiente de la ciudad?:
0

Cómo valora la comida de la ciudad?:
0

Cómo valora la cultura de la ciudad?:
0

Cómo valora el clima de la ciudad?:
0

Cómo valora el tamaño de la ciudad?:
0

Valorar

Ilustración 13: Pantalla para puntuar las ciudades

Añadir_valoracionesBE.php

Como su nombre indica este script funcionará a modo de backend, su funcionalidad consistirá en insertar las valoraciones realizadas por los usuarios.

En primer lugar recuperará todos los parámetros que hayan sido pasados desde el script anterior a través del método post, para obtener así las valoraciones que ha realizado el usuario, en segundo lugar recuperará el id de la ciudad y el email del usuario que están almacenados, a continuación, realizará una consulta sobre la tabla “valoraciones” con el fin de comprobar si ya existe una valoración por parte de ese usuario sobre esa ciudad, en caso de que no exista se creará un registro en la base de datos con el email del usuario y el id de la ciudad, y por último, existiese o no el registro realizará un update sobre la tabla valoraciones para la combinación del email y el id de la ciudad dando valor a los atributos, esto permite que los usuarios también puedan modificar sus valoraciones si así lo desean.

Acceso_chat.php

El siguiente script da acceso a la última funcionalidad del servicio web descrita al inicio de este documento, está funcionalidad consiste en ofrecer al usuario una serie de chats para que se pueda comunicar con otros usuarios.

En esta pantalla los usuarios van a ser bienvenidos al chat y se les va a permitir seleccionar a que chat quieren acceder, entre los chats a los que se puede acceder encontramos a los que fue agregado en el momento de registro en la página web de manera automática, estos son los chats con usuarios que provienen de la misma universidad en su país de origen, usuarios que van a la misma universidad en su experiencia erasmus y un chat con todos los usuarios que van a la misma ciudad en su experiencia erasmus, por otro lado, hay una

cuarta opción que es el acceso directo a chat privados que permiten contactar individualmente con otros usuarios. Los tres primeros botones redirigen al mismo script denominado “*devolver_mensajes.php*” al que se le añade un parámetro que indica a que chat se está queriendo acceder según el botón que pulsen, en el caso del botón de acceso a los chats privados, este redirige a otro script denominado “*chats_privados.php*” para la gestión de los chats privados del usuario.

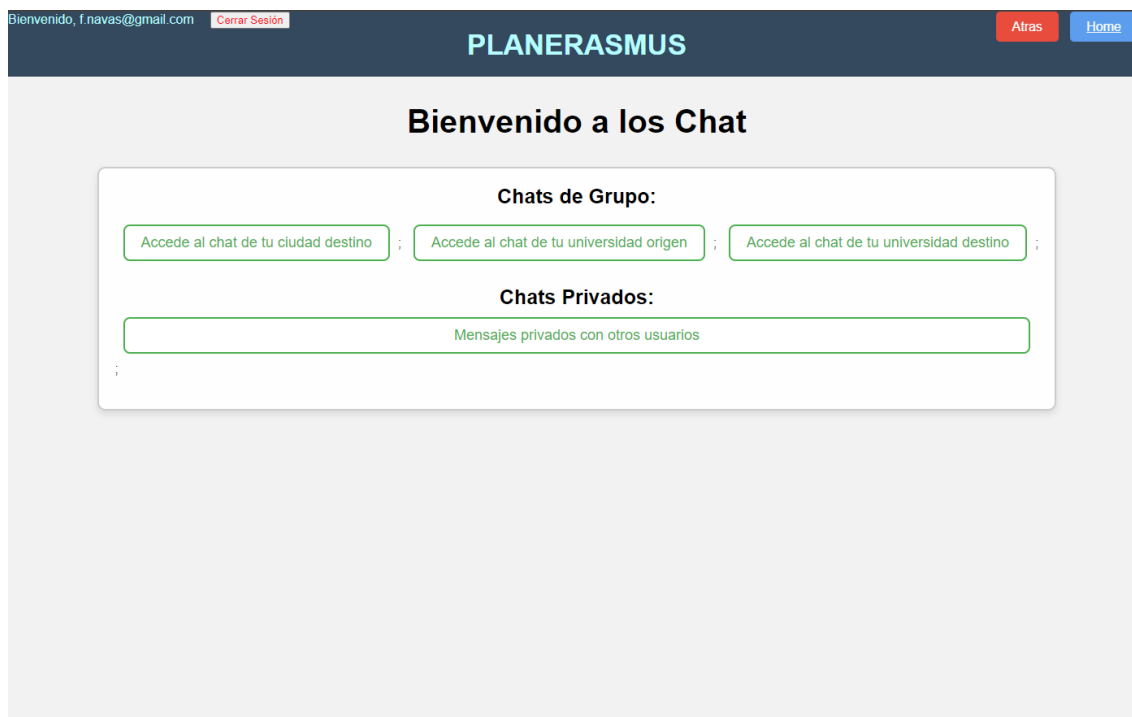


Ilustración 14: Pantalla para acceder a los chats

Devolver_mensajes.php

El siguiente script mostrará los chats del usuario, lo primero que hará será recuperar el parámetro pasado en la url para comprobar el valor de este, en caso de que se corresponda con que es de tipo ciudad o universidad se realizará una consulta realizando un inner join entre las tablas “chat”, “pertenece” y “referente” para poder recuperar el id del chat, uniendo las tablas chat y pertenece a través del id del chat, y se unirán las tablas chat y referente también a través del id del chat, se filtrará por el email del usuario con el fin de recuperar solo los chats a los que pertenece el usuario y el tipo de referencia, es decir, a cual de los chats se refiere si al de ciudad o el de universidad, en el caso de las universidades también se filtrarán por el tipo de universidad, la origen o la destino. En el caso de encontrarnos ante un chat privado el valor del parámetro pasado desde el script previo contendrá ya el id del chat.

Una vez obtenido el id del chat se recuperarán los mensajes de este chat y se imprimirán en pantalla para la visualización por parte del usuario.

El script también cuenta con una barra para poder escribir mensajes en los chats, esta barra será un formulario que tras escribir el mensaje realizará una llamada al script “*introducir_mensaje.php*” con un método post.

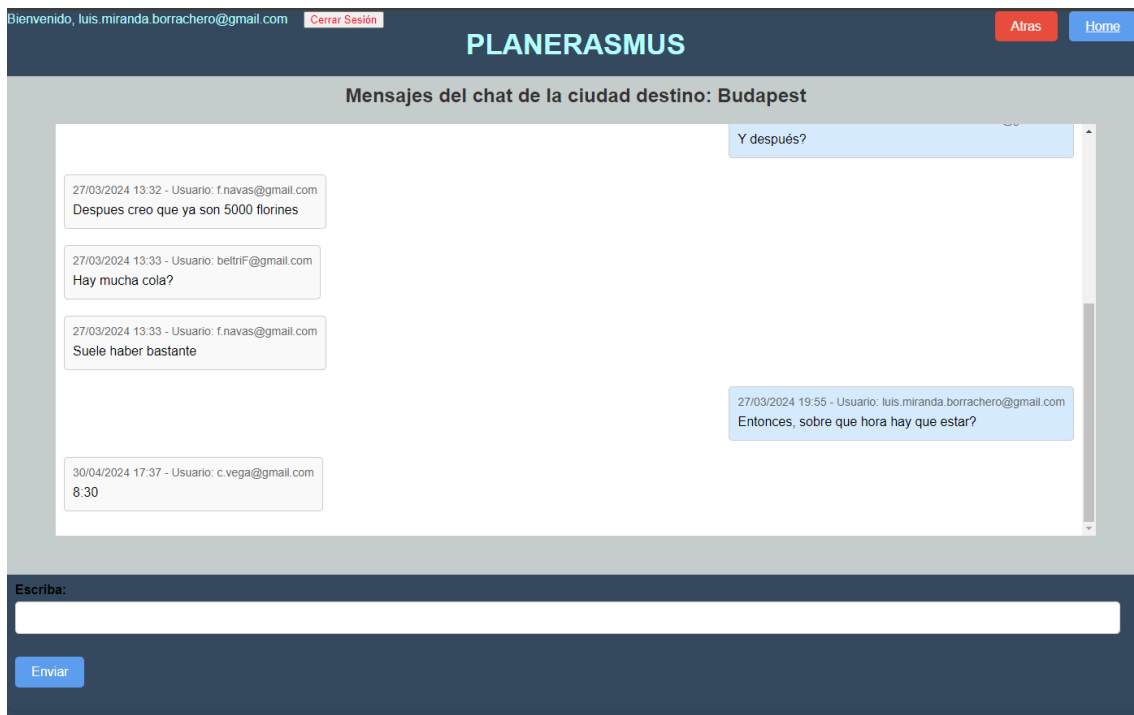


Ilustración 15: Visualización de los chats

Introducir_mensaje.php

El siguiente script funcionará a modo de backend y su funcionalidad consiste en introducir el mensaje escrito por el usuario, en la base de datos.

Para ello, lo primero que hará será recuperar los parámetros del método post, que recogen el id del chat y el texto del mensaje que se quiere introducir. A continuación, realizará una inserción en la tabla “mensaje” con el mensaje, el email del usuario que lo ha escrito y el identificador del chat.

Chats_privados.php

El siguiente script tendrá la funcionalidad de dar acceso a los usuarios a sus chats privados.

En este script se va a dar al usuario una serie de opciones, las primeras van a ser con el fin de buscar usuarios dentro del servicio web, tendrá hasta cuatro formas distintas para buscar usuarios, la primera consistirá en una búsqueda por nombre o email, en esta opción el usuario podrá escribir el nombre o el email del usuario, en la siguiente opción el usuario podrá filtrar a los usuarios por la ciudad donde se encuentran de erasmus, en la siguiente podrá filtrar por la universidad origen del usuario, y en la última opción por la universidad destino del mismo. Habrá otra opción adicional que permitirá al usuario acceder directamente a los chats ya creado previamente por el usuario al contactar con otros usuarios.

Para realizar esto, las cuatro primeras opciones serán formularios que realizarán una llamada al script “*buscar_usuario.php*” a través de un método post, pasándole los parámetros correspondientes para cada caso. La última opción realizará una llamada al script “*explorar_chats.php*”.

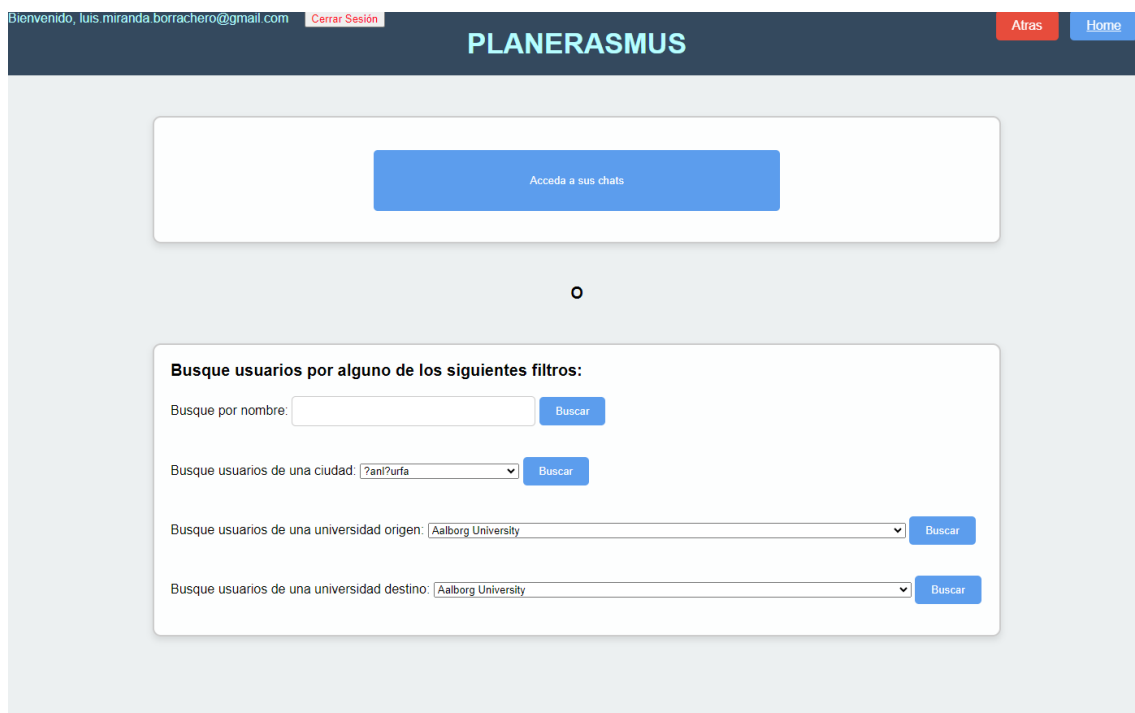


Ilustración 16: Pantalla para búsqueda de usuarios, o el acceso a chats

Explorar_chats.php

El siguiente script permitirá al usuario acceder a sus chats privados ya establecidos, este script es accedido desde el script *“chats_privados.php”* y se accede a través de la última opción mencionada en la descripción de este.

Lo primero que realiza es la recuperación del email del usuario que tiene iniciada la sesión, acto seguido se realizará una consulta que devuelva los chats privados a los que pertenece el usuario, para poder imprimir todos sus chats lo que se hará será un bucle sobre los resultados, se hará una consulta para conocer los datos del usuario con quien tiene establecido el chat y se imprimirá su email, su nombre, así como un botón para acceder al chat con dicho usuario.

Para realizar todo esto, la primera consulta mencionada se realizará entre las tablas *“chat”* y *“pertenece”* a través de un inner join que las unirá respecto al id del chat, filtrando porque sea privado y porque el valor del email del usuario de la tabla *“pertenece”* corresponda con el usuario en cuestión.

Para recuperar los datos de los usuarios con los que se comparte el chat se realizará otra consulta, en este caso entre la tabla *“pertenece”* y la tabla *“usuario”* comparando el email de este en ambas tablas, y filtrando por los id del chat recuperados en la consulta anterior e indicando que el valor del email del usuario no puede corresponder con el del usuario que accede a sus chats.

Finalmente, el botón para acceder al chat realizará una llamada al script *“devolver_mensajes.php”* indicando por parámetro el id del chat.

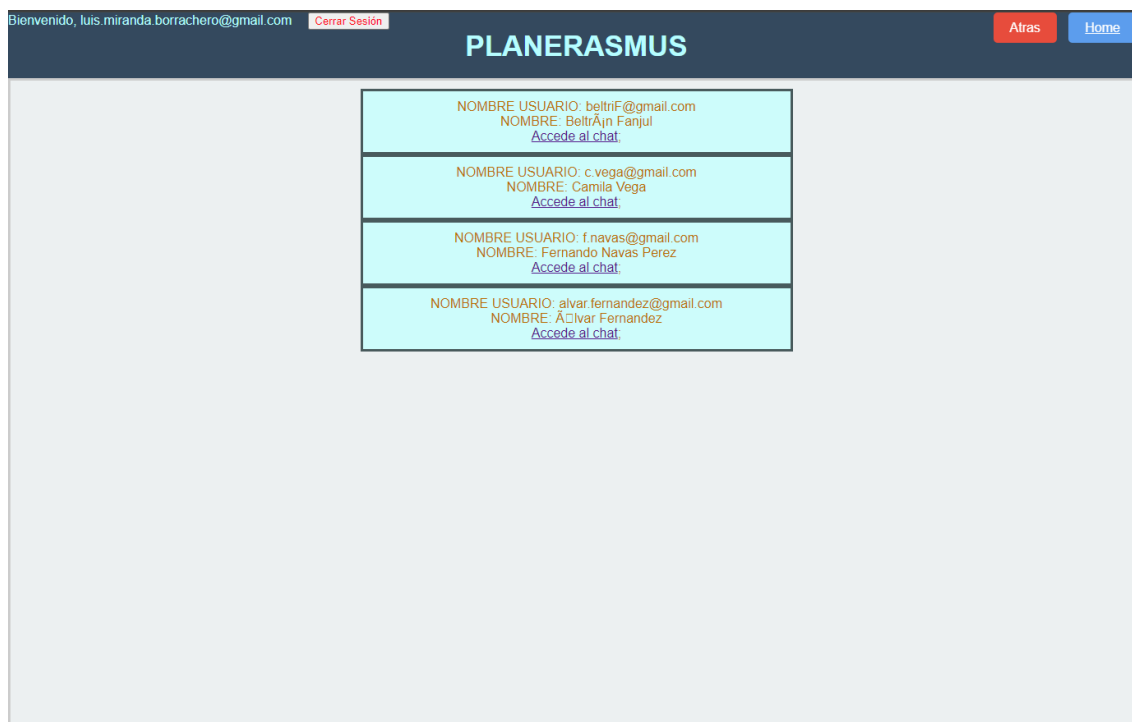


Ilustración 17: Pantalla para ver los chats privados

Buscar_usuario.php

El siguiente script imprimirá una lista de usuarios según los filtros establecidos por parte del usuario con el fin de poder contactar con un usuario que cumpla dichas características.

Para realizar esto lo primero que se hace es recuperar los filtros pasado por el usuario, cabe destacar que hay cuatro tipos distintos de filtros, se tratará de recuperar todos, pero solo se podrá recuperar el filtro seleccionado por el usuario. Tras esto dependiendo del filtro usado se realizará una consulta u otra, si es por nombre se recuperarán los usuarios y su ciudad aplicando un filtro en el que se aplicará que el nombre y/o el email contengan el patrón de texto establecido por el usuario, si es por la ciudad, se aplicará el filtro sobre la id de la ciudad y será el mismo procedimiento para la universidad. Cabe destacar que para todas las consultas se aplicará un filtro para que no se pueda devolver al propio usuario que está buscando otros usuarios.

Una vez recuperado los usuarios que cumplen los filtros, así como su cruce con la tabla "ciudad" se imprimirá cada usuario en un cuadro, imprimiendo su email, su nombre y su ciudad destino, así como un botón con el texto "Escribir" que realiza una llamada al script "*crear_chat.php*" pasando como parámetro el email del usuario correspondiente.

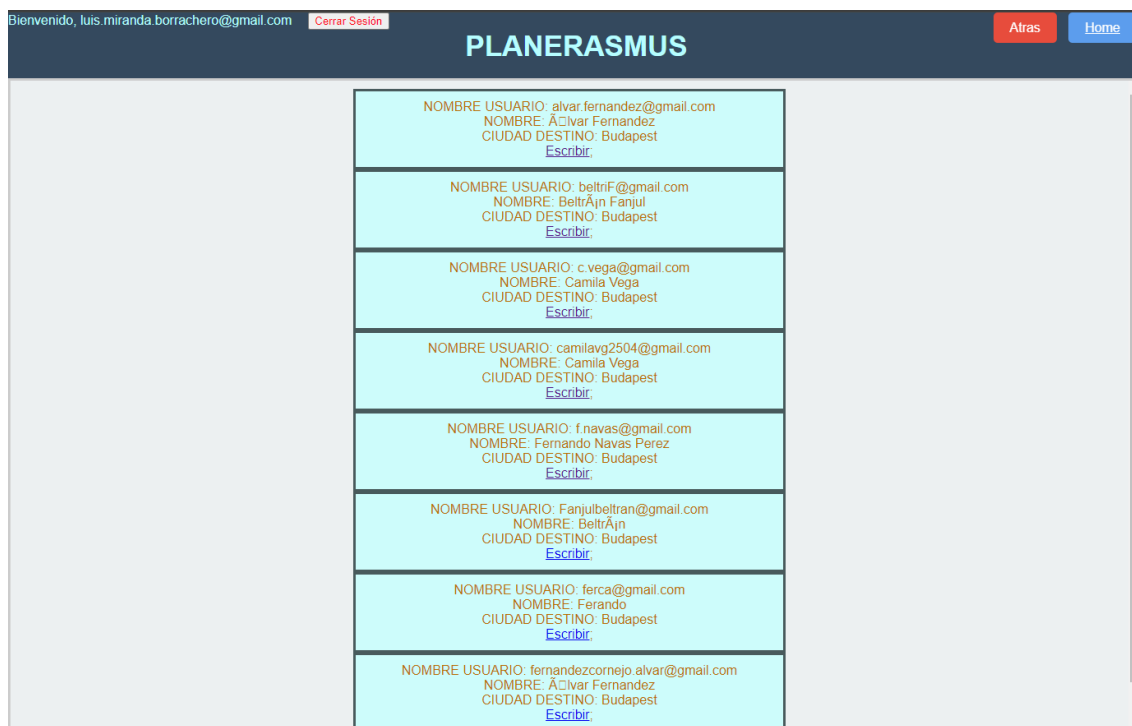


Ilustración 18: Pantalla para visualizar los resultados de una búsqueda

Crear_chat.php

El siguiente script funcionará exclusivamente a modo de backend, su funcionalidad consistirá en crear un chat entre dos usuarios de la aplicación, esto se debe a que en un inicio no existen chats entre cada uno de los usuarios por la cantidad de registros que supondría, por ello, tan solo se crea el chat cuando se quiere hablar con el usuario en cuestión.

Este script es ejecutado tras una llamada al script anterior, los usuarios pueden filtrar en dicho script una búsqueda de otros usuarios de la aplicación y entre los resultados pueden aparecer usuarios con los que ya se hayan creado los chats, por ello, lo primero que realizará este script es recuperar tanto el email del usuario al que se quiere seguir como el email del usuario que tiene iniciada la sesión, una vez hecho esto se recuperarán todos los chats a los que pertenece el usuario que está haciendo la solicitud, filtrando para que la consulta devuelva solo los chats que son privados y no devuelva los chats de grupo a los que pertenece, se realizará lo mismo para el otro usuario y se comparará si pertenecen a un chat común.

Para realizar esto se realiza una consulta con un inner join entre las tablas “chats” y “pertenece”, uniéndolas a través del id del chat y filtrando por el email del usuario, así como que el atributo booleano de la tabla “chats” denominado “privado” sea true. A continuación, se guardarán en un array todos los id de los chats recuperados haciendo uso de un bucle while.

Se realizará la misma consulta previa pero ahora para recuperar los chats a los que pertenece el usuario con el que se quiere establecer una conversación, una vez recuperados dichos chats lo que se realizará es un bucle sobre los chats recuperados, en los que en cada paso se comparará el id del chat con todos los id de los chats almacenados en el array mencionado previamente, para hacer

esto se realizará un bucle for dentro de un bucle while, en caso de que los id coincidan se establecerá una variable booleana como true, que permitirá la salida tanto del bucle for como del bucle while.

Para finalizar, se estudiará el valor de la variable boolean, en caso de que sea true es que ya existe un chat entre los dos usuarios por lo que se realizará una llamada al script “*devolverMensajes.php*” pasándole como parámetro el id del chat. En caso de que sea false, no existirá ningún chat entre dichos usuarios, por lo que se procederá a crear dicho chat, para ello se creará un nuevo registro en la tabla “chat”, marcando como true el atributo “privado”, y acto seguido se introducirá en la tabla “pertenece” un registro con el email del usuario que tiene iniciada la sesión y el id del chat recientemente creado y otro registro con el id del chat y el usuario que se trata de contestar, finalmente se realiza la misma llamada al script “*devolverMensajes.php*” con el id del chat recientemente creado.

Elementos comunes en los scripts

Los scripts descritos previamente cuentan además con una serie de elementos comunes para todos ellos, que se resumen a continuación:

- Elementos visuales:

- ◆ Todos los scripts van a contar con una cabecera común que se puede apreciar en las imágenes que acompañan a las descripciones. Esta cabecera cuenta con un fondo azul marino, ocupando 80px de la altura de la página y ocupando el total de la anchura de esta. Dentro de este espacio podemos encontrar, en el caso de que exista una sesión activa, un mensaje de bienvenida, junto con el nombre de usuario y junto con un botón que indica “cerrar sesión” y que ejecutará el script encargado de cerrar la sesión, en caso de que no exista ninguna sesión activa aparecerá tan solo el botón con el texto “iniciar sesión” que nos permitirá acceder al script para iniciar sesión. Y en el centro se va a encontrar el nombre de la aplicación.
- ◆ A la derecha de dicha cabecera vamos a encontrar dos botones de navegación, el primero de ellos nos va a permitir volver a la página principal conocida como “home”, el segundo de ellos nos permite volver al script que se ha ejecutado previamente, cabe destacar que si el script ejecutado previamente es de tipo backend este botón no nos hará navegar a otro script visto visualmente.



Ilustración 19: Cabecera común

- **Elementos funcionales:** Para la realización de lo dicho previamente, así como para la gestión de las sesiones todos los scripts cuentan con un código al inicio que se encarga de la gestión.
 - ◆ Para la gestión de las sesiones se va a hacer uso del manejo de las cookies de sesión, estas cookies permiten almacenar

variables de una sesión de un script a otro, para ello al comienzo de cada script se realiza una función denominada “sesión_start” que permite la creación de una sesión en esa script.

- ◆ Para gestionar si un usuario ha iniciado sesión o no se hace uso de una variable de sesión denominada “usuario”, que se recupera y en caso de que no exista se imprime el botón “cerrar sesión” y en caso de que exista se imprime el mensaje de bienvenida con el correo del usuario almacenado en ella.
- ◆ Para la navegación existe otra variable de sesión denominada “anterior”, lo que se hará será recuperar el valor de dicha variable con el fin de conocer el valor de la url del script anterior y se cambiará el valor con el nombre de la script actual, para que cuando se dé al botón “anterior” se le redirija al valor obtenido de esa variable.

Inserción de datos iniciales

La mayoría de los datos generados y guardados en la base de datos son generados por parte de los usuarios y, por lo tanto, la gestión de la introducción de estos en la base de datos es realizado por parte de los scripts descritos en el capítulo anterior.

No obstante, hay una serie de datos que han de ser introducidos inicialmente y por parte de los desarrolladores del servicio, en este caso por mí. Estos datos son básicamente los referentes a las ciudades europeas y a las universidades mismas, que quedan a disposición del usuario para que seleccione donde va a ir, de donde viene, para así poder acceder a sus chats, o para que puedan seleccionar la ciudad sobre la cual quieren obtener información y/o valorar, etc.

Obtención de los datos

Los datos a buscar han sido principalmente las universidades europeas junto con las ciudades a las que perteneces, para poder encontrar dichos datos se ha hecho uso de los ranking europeos de las mejores universidades de Europa ya que era la manera más sencilla de poder obtener el máximo número posible de universidades, finalmente los datos han sido obtenido de la página web: <https://www.topuniversities.com/europe-university-rankings> , donde se ha podido recuperar a través de un formato Excel un ranking con 644 universidades, evidentemente estas no son todas las universidades de Europa pero si un gran número de estas, el resto de universidades podrían ser añadidas via petición de los usuarios que no encuentren entre las 644 universidades su universidad origen y/o destino.

Esto nos ha ofrecido las universidades, pero no se ha podido obtener un listado con las universidades y la ciudad a la que pertenece, por ello se ha hecho uso de IA, en este caso Chat GPT para que pasado el listado indicará la ciudad a la que pertenece dichas universidades, obteniendo registros como los mostrados a continuación:

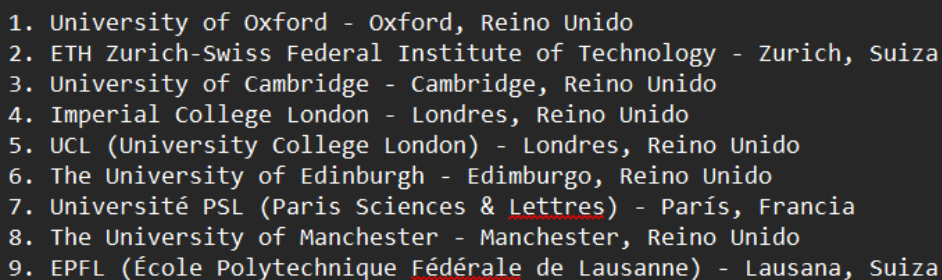
- 
- A screenshot of a list of universities and their corresponding cities. The list is numbered from 1 to 9. The text is displayed in a monospaced font on a dark background. The items are: 1. University of Oxford - Oxford, Reino Unido; 2. ETH Zurich-Swiss Federal Institute of Technology - Zurich, Suiza; 3. University of Cambridge - Cambridge, Reino Unido; 4. Imperial College London - Londres, Reino Unido; 5. UCL (University College London) - Londres, Reino Unido; 6. The University of Edinburgh - Edimburgo, Reino Unido; 7. Université PSL (Paris Sciences & Lettres) - París, Francia; 8. The University of Manchester - Manchester, Reino Unido; 9. EPFL (École Polytechnique Fédérale de Lausanne) - Lausana, Suiza.
1. University of Oxford - Oxford, Reino Unido
 2. ETH Zurich-Swiss Federal Institute of Technology - Zurich, Suiza
 3. University of Cambridge - Cambridge, Reino Unido
 4. Imperial College London - Londres, Reino Unido
 5. UCL (University College London) - Londres, Reino Unido
 6. The University of Edinburgh - Edimburgo, Reino Unido
 7. Université PSL (Paris Sciences & Lettres) - París, Francia
 8. The University of Manchester - Manchester, Reino Unido
 9. EPFL (École Polytechnique Fédérale de Lausanne) - Lausana, Suiza

Ilustración 20: Datos iniciales de universidades y ciudades

Manejo e introducción de los datos

Tras la obtención de una lista con las universidades y las ciudades, pasamos al manejo de estas y la posterior introducción de los datos en la base de datos. Para poder realizar esto se ha programado un script en lenguaje java.

Este script ha seguido varios pasos, el primero de todos ha consistido en hacer uso de una función que he denominado “ejecutar” y que abriendo el archivo

mostrado en la ilustración 20 a dado paso a generar otro archivo quedándose solo con las ciudades europeas:

```
Oxford Reino Unido
Zurich Suiza
Cambridge Reino Unido
Londres Reino Unido
Londres Reino Unido
Edimburgo Reino Unido
París Francia
Manchester Reino Unido
Lausana Suiza
Londres Reino Unido
Múnich Alemania
Londres Reino Unido
Delft Países Bajos
```

Ilustración 21: Archivo Ciudades Europeas

No obstante, como también se puede observar en el ejemplo, en el ranking se da el caso de que haya universidades pertenecientes a este que sean de la misma ciudad como es lógico, lo que genera el problema de tener duplicidades de ciudades en este archivo, por ello, el siguiente paso ha sido la ejecución de otra función del programa que ha recibido el nombre de “comprimir”, sobre el archivo anterior para la obtención de un segundo archivo con tan solo un registro de cada ciudad, como se muestra a continuación:

```
Oxford,Reino Unido
Zurich,Suiza
Cambridge,Reino Unido
Londres,Reino Unido
Edimburgo,Reino Unido
París,Francia
Manchester,Reino Unido
Lausana,Suiza
Múnich,Alemania
Delft,Países Bajos
```

Ilustración 22: Archvo con ciudades comprimidas

Una vez habiendo moldeado los datos como era necesario se ha pasado a la inserción de los datos en la base de datos, para ello se han realizando varias funciones que con el uso de jdbc conectaban con la base de datos para realizar los diferentes insert.

En primer lugar, se ha ejecutado la función “insertarCiudades” que ha sido la encargada de introducir los datos de las ciudades haciendo uso del archivo generado en el paso previo, era necesario primero introducir las ciudades ya que las universidades tienen una relación con estas y heredan de las ciudades una clave foránea que queda registrada en sus registros.

En segundo lugar, se han introducido los datos de las universidades en la base de datos a través de otra función denominada “insertarUniversidades” que

primero recuperaba el id de la ciudad a la que pertenecía para luego introducir la universidad en el sistema.

Y como último paso, se han creado los registros de los chats globales de las ciudades y de las universidades, que son creados desde el inicio de la aplicación y que necesitaban que existieran los datos de las ciudades y las universidades ya que son referentes a ellos, esto se ha hecho a través de dos funciones denominadas “crearChatsUniversidades” y “crearChatsCiudades”.

Test de usabilidad

Un paso fundamental a la hora de diseñar de una aplicación web o móvil es evaluar la usabilidad que esta tiene, valorando si es fácil de manejar, es intuitiva y clara. Por ello, en el siguiente apartado se realizará un pequeño estudio con usuarios de prueba donde se les requerirá realizar una serie de acciones y se valorará la realización de estas.

Diseño de las acciones a realizar

Se les va a pedir a los usuarios que realicen las siguientes acciones:

1. **Registrarse en la aplicación:** Lo primero que se les requerirá es que procedan a registrarse en la aplicación, se les pedirá a todos los usuarios que al registrarse indiquen que su universidad origen es la UPM y su universidad destino la Budapest University of Technology and Economics, principalmente porque no todas las universidades tienen procesos creados, o pruebas realizadas en sus chats. El objetivo para los usuarios es acceder en 2 clicks a la pantalla de registro y no tener problemas para registrarse en la aplicación.
2. **Marcar como completado un proceso:** La siguiente acción consistirá en pedirle al usuario que acceda a los procesos que tiene que realizar de su universidad origen y marquen como completado la entrega de learning agreement. El objetivo para el usuario es acceder a los procesos en 2 clicks y un tercero para marcarlo como completado.
3. **Consultar información sobre dónde comer en Budapest:** Para realizar esta acción se deberá acceder a los foros de la aplicación y entrar en el foro de comida de Budapest. Se necesitarán 5 clicks para completar la acción.
4. **Escribir un mensaje a otro usuario:** Se le pedirá al usuario que busque un usuario que vaya a realizar su Erasmus en Budapest. Para ello deberá hacer una búsqueda de usuarios filtrando por ciudad destino. Se requerirán hasta 8 clicks para enviar el mensaje
5. **Puntuar la última ciudad en la que estuvo:** Se pedirá al usuario que valore la última ciudad europea en la que ha estado. Se requerirán 5 clicks para acceder al formulario de las valoraciones.

Resultados

A continuación, se recogen las notas escritas sobre los resultados de las pruebas.

Usuario 1:

- Características: Mujer, 24 años.
- Ha estado de Erasmus: Sí.

Registrarse:

Ha accedido en el número de clicks deseados. No ha tenido grandes dificultades, solo mucha dificultad en encontrar las universidades en la lista. ¿Muy pequeño?

Marcar un proceso:

Ha accedido en el número de clicks correcto. Ninguna dificultad.

Mejorar la visualización

Consultar info de comida en Bdp:

Ninguna dificultad ha accedido a los foros sin problema. Otra vez la lista poco usable.

Escribir un mensaje a otro usuario:

Algo confuso el elegir chats, se ha ido primero al chat de la universidad

La pantalla de buscar confusa no sabe si tiene que rellenar todo para buscar, ha dudado

Otra vez las listas, sin ratón se maneja muy mal

Puntuación de una ciudad:

Ha accedido correctamente

Surge una problemática que no saben cómo valorar. Por ejemplo, el coste de vida no sabía si poner una nota alta significa que era muy caro o si, como debe ser, valoras tu dándole una nota, es decir si para ti ha sido barato le das buena nota

Notas extras:

Está muy intuitivo, uso fácil.

Usuario 2:

- Características: Hombre, 22 años.
- Ha estado de Erasmus: Sí.

Registrarse:

Lo consigue en los clicks correctos, se queja de que ni le devuelve al inicio y que el nombre no especifica que oner.

Proceso en curso:

Todo bien

Acceso a foro:

Ha accedido correctamente, he realizado un comentario. Debería volver al foro más que atrás.

Hablar con un usuario:

Primero, diferenciar claramente el acceso a los chats privados y a los de grupo, se pierden los usuarios.

Segundo, en la búsqueda de chat hay que hacer un cambio ya que al haber un botón abajo que da acceso a los chats ya creados, el usuario se ha pensado que debería rellenar todos los cambios

Valoraciones:

Vuelta a valoraciones de la ciudad. 2 dígitos de decimales como mucho

Notas extra:

Diseño de la pantalla de inicio cambiar, meter una cabecera igual en todas las páginas con el nombre de la página web.

Caracteres raros hay que eliminarlos.

Usuario 3:

- Características: Hombre, 21 años
- Ha estado de Erasmus: Sí

Registrarse:

Todo correcto

Procesos en curso:

Todo correcto

Acceso a foro:

Todo correcto

Hablar con otro usuario:

Se mete primero en los chats de grupo, luego pasa lo mismo que con otros usuarios, trata de filtrar por todas las cosas que hay.

Valoraciones:

Todo correcto

Notas extra:

El botón de navegación “atrás” no funciona siempre, y a veces no realiza ninguna acción.

Usuario 4:

- Características: Hombre, 22 años
- Ha estado de Erasmus: No

Registrarse:

Todo bien, aunque ha escrito el email y contraseña y ha dado a registrar

Procesos en cursos:

Realizado con éxito:

Acceso a foro:

Ha accedido a valoraciones y ha obtenido la puntuación, no es realmente erróneo, luego ha ido al foro y lo ha hecho bien

Hablar con otro usuario:

Lo ha hecho bien, aunque ha dudado bastante

Valoraciones:

Lo ha hecho bien

Nota extra:

Nada que comentar

Usuario 5:

- Características: Hombre, 22 años
- Ha estado de Erasmus: Sí

Registrarse:

Todo bien, aun que le ha costado encontrar las universidades.

Procesos en curso:

Todo bien

Acceso a foro:

Bien, aunque al igual que otro usuario, entró primero en valoraciones para buscar la información de como se valora la ciudad, debería ser más específico en qué se requiere, o los usuarios no asocian directamente los foros con los comentarios de la gente.

Hablar con otro usuario:

Se ha perdido en la búsqueda por ciudad, no es muy intuitivo según él, mismo problema que otros usuarios.

Valoraciones:

Lo ha hecho bien

Notas extra:

No funciona el botón de “atrás”.

Análisis de los resultados.

Tras la realización de las pruebas de usabilidad se ha podido observar como hay ciertos elementos que no funcionan bien o que no son intuitivos para los usuarios.

Errores de usabilidad:

1. El principal fallo que han tenido los usuarios se ha producido en la búsqueda de un usuario que resida en Budapest para escribirle un mensaje directo.
En una primera instancia el acceso a los chats privados no queda tan claro, varios usuarios han tratado de acceder a los chats de grupo para buscar un usuario.

La mayor problemática, sin embargo, ha residido en cómo los usuarios debían buscar por ciudad, esto se debía a la disposición de los filtros que invitaban a los usuarios a pensar que debían rellenar todos los campos para poder hacer el registro.

2. Otra acción que ha sucedido con varios usuarios es como han accedido a las valoraciones para obtener información sobre la comida, ya que el usuario no asocia los foros con la búsqueda de información.
3. Las listas de ciudades o universidades no han sido muy cómodas para ciertos usuarios.
4. Algunos usuarios comentan que estaría bien mejorar la visualización de ciertas páginas.

Errores funcionales:

1. El principal error funcional que han reportado los usuarios es el funcionamiento del botón de “Atrás” que es cierto que funciona algo regular.
2. Hay varios errores que tienen que ver con a donde te lleva la página sobre todo al realizar una acción, para los usuarios no es comprensible porque cuando te registras te devuelve a la pantalla de registro en vez a la de inicio, o cuando realizas un comentario en el foro que no vuelve al foro sino a la selección de la ciudad, o cuando realizas una valoración que no vuelve a las valoraciones sino a seleccionar la ciudad.

Cambios realizados

Se han realizado cambios con el fin de poder mejorar la usabilidad de los usuarios y de corregir los errores funcionales de la página web.

Cambios de usabilidad:

Se han realizado varios cambios de usabilidad.



Ilustración 23: Cambios en el acceso a chats

El primero para que el usuario tenga claro la diferencia entre los chats privados y los chats de grupo se ha introducido en el acceso a estos un mensaje indicando cómo acceder a cada uno, como se puede ver en la ilustración 23. De esta manera se espera reducir sus errores a la hora de acceder a los chats privados.

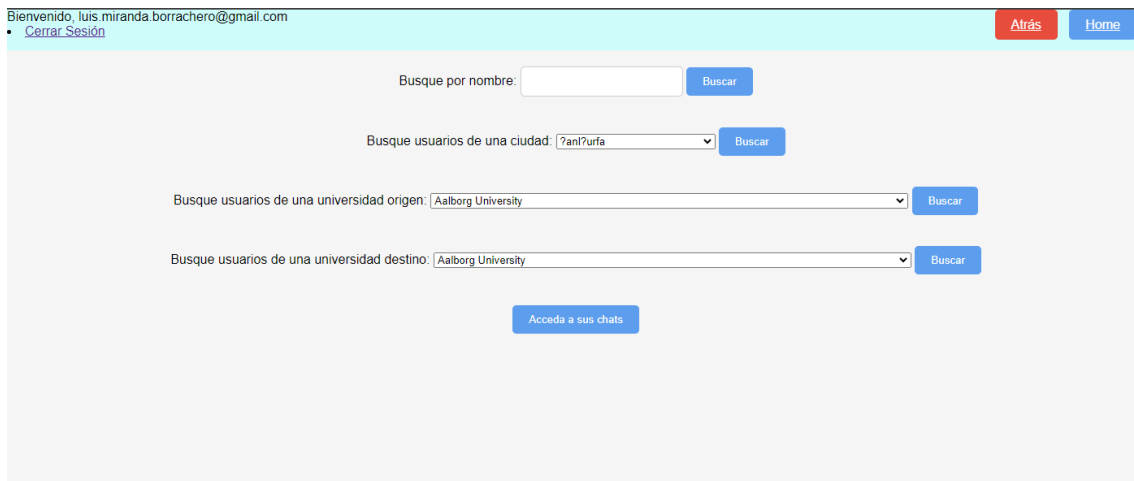


Ilustración 24: Antigua visualización del acceso a chats privados

En la ilustración se puede observar cómo era la antigua interfaz de los chats privados y se puede entender el motivo de que los usuarios cometieran tantos fallos.

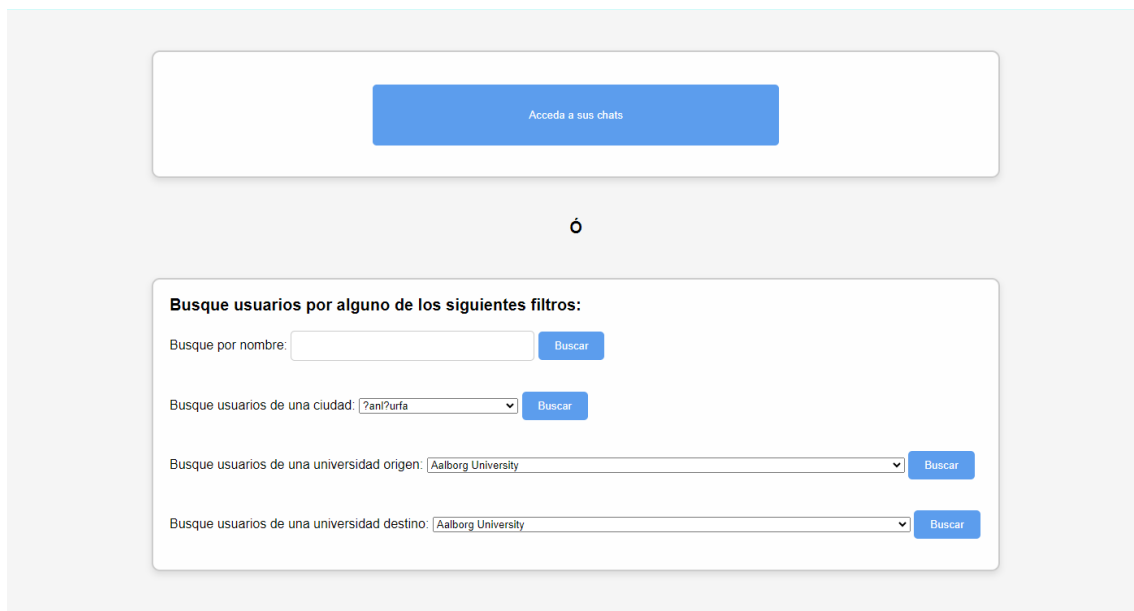


Ilustración 25: Cambio en la búsqueda de chats privados

El error más común era que en la búsqueda de usuarios para escribirles un mensaje privado, la mayoría de estos al ver al final el botón de acceder a los chats, se pensaban que debían rellenar todos los filtros y después darle ahí, ahora se ha dividido claramente el acceso a los chats privados de la búsqueda de usuarios, y se ha dejado más claro como buscar. Esto se ve reflejado en la ilustración 25.

Cambios funcionales:

En primer lugar, se han tratado de arreglar las problemáticas a nivel funcional de la aplicación. El error principal que se reportaba por los usuarios y que ciertamente no estaba funcionando correctamente tenía que ver con el botón de atrás, el cual como se explicó previamente trataba de regresar a la url anterior

que se había almacenado en una variable de sesión, el problema residía en que al haber varios script cuyo funcionamiento es tan solo a nivel de backend, cuando estos se ejecutaban la url cambiaba y dejaba de apuntar al script correcto, además, solo se podía volver una página para atrás porque solo se almacenaba dicha página, para poder arreglar este problema se ha insertado un script de java con una función denominada volveratras que lo que realiza es una acción “window.history.back” que permite a volver a la página anterior según el historial, además, esto permite volver para atrás todos los script que se considere.

Por otro lado, otro de los problemas era el volver a la página que correspondía tras realizar una acción, como volver al foro tras comentar o volver a la pantalla de inicio tras registrarse, todos estos errores han sido corregidos.

Cambios visuales:

Varios usuarios han reportado que ciertas pestañas podían ser mejor visualmente hablando, por ello, se ha hecho un rediseño en la visualización de las pestañas.

Las muestras mostradas a lo largo de la memoria corresponden con la nueva versión, no obstante, la versión mostrada a los usuarios no correspondía con dicha versión.

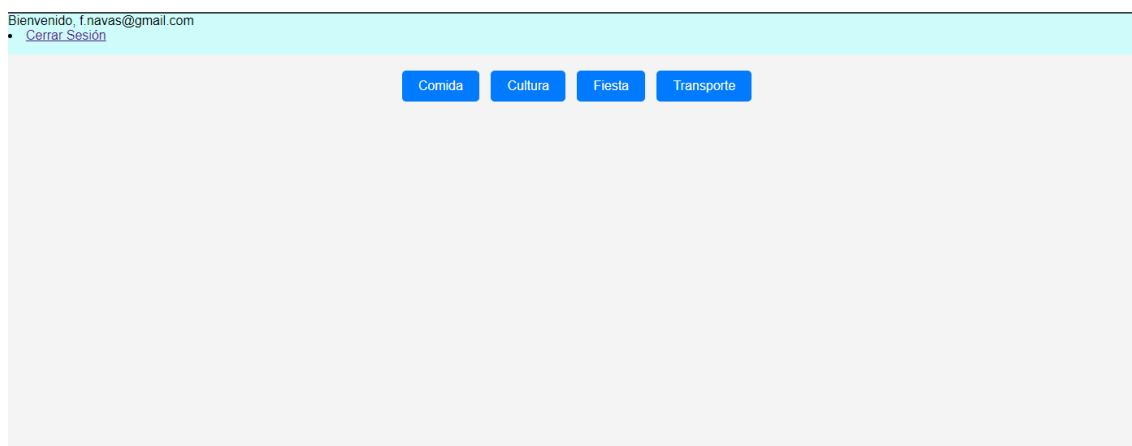


Ilustración 26: Versión antigua

En la ilustración 25 se puede observar un ejemplo de la antigua versión de la página web, en ella se tenía un header con un azul turquesa, y un fondo grisáceo, el contenido en ciertas páginas era muy pobre y soso como el que se muestra en el ejemplo.

Por ello, se rediseñó el header, cambiando el color, incluyendo en todas las páginas los botones de navegación e incluyendo el título de la página web a modo corporativo.

Conclusión

El diseño y desarrollo de una página web dedicada a ofrecer un servicio integral para estudiantes Erasmus representa un avance significativo en la gestión y mejora de su experiencia académica y personal en el extranjero. A través de este proyecto, se han implementado diversas funcionalidades que facilitan tanto los procesos burocráticos como la integración social y cultural de los estudiantes.

1. Gestión de Procesos Burocráticos:

La primera funcionalidad de la página web se centra en simplificar los trámites administrativos que los estudiantes deben realizar antes y durante su estancia Erasmus. La creación de un portal donde las universidades de origen y destino pueden publicar los procesos abiertos permite a los estudiantes tener una visión clara de las tareas pendientes, con descripciones detalladas y fechas límite. La opción de marcar procesos como completados ayuda a los estudiantes a mantener un seguimiento organizado de sus obligaciones, funcionando también como una lista de verificación eficiente.

2. Conexión entre Estudiantes:

El segundo apartado busca fomentar la colaboración y el apoyo mutuo entre los estudiantes Erasmus. Mediante la creación de chats grupales basados en la universidad de origen, la universidad de destino y la ciudad de destino, los estudiantes pueden compartir experiencias, consejos y resolver dudas. Esta funcionalidad no solo facilita la preparación previa a la experiencia Erasmus, sino que también proporciona un soporte continuo durante la estancia. La opción de chats privados, con filtros específicos para buscar a otros estudiantes, amplía aún más las posibilidades de interacción y networking.

3. Información Relevante sobre Ciudades Europeas:

La tercera funcionalidad ofrece un foro estructurado en subforos dedicados a diferentes aspectos de las ciudades europeas. Este espacio permite a los estudiantes compartir y obtener información sobre cultura, gastronomía, ocio y transporte, basándose en experiencias personales. Los comentarios y valoraciones de otros estudiantes proporcionan una fuente valiosa de conocimiento práctico, ayudando tanto en la elección de la ciudad de destino como en la planificación de viajes.

4. Portal de Valoraciones:

Complementando la información del foro, el portal de valoraciones permite a los estudiantes puntuar distintos aspectos de las ciudades. Esta herramienta de retroalimentación es crucial para futuros estudiantes, quienes pueden tomar decisiones informadas basándose en las opiniones y experiencias de sus predecesores.

Tecnologías Utilizadas:

El desarrollo de esta página web ha involucrado el uso de una base de datos relacional y el aprendizaje y aplicación de lenguajes de programación como PHP, HTML y CSS. Estos conocimientos técnicos han sido fundamentales para construir una plataforma funcional, intuitiva y segura.

Conclusión final:

En resumen, esta página web no solo aborda las necesidades administrativas de los estudiantes Erasmus, sino que también promueve una comunidad de apoyo y colaboración. La combinación de funcionalidades prácticas y sociales asegura que los estudiantes estén bien preparados y acompañados durante toda su experiencia. Este proyecto demuestra cómo la tecnología puede ser utilizada para mejorar la calidad de vida de los estudiantes en movilidad internacional, facilitando su adaptación y éxito en un entorno nuevo.

Anexos

SCRIPT Creación BBDD:

```

Create table ciudad(
  id_ciudad int auto_increment NOT NULL,
  nombre varchar(500) NOT NULL,
  pais varchar(50) NOT NULL,
  primary key (id_ciudad)
);
Create table universidad(
  id_universidad int auto_increment,
  nombre varchar (100),
  id_ciudad int NOT NULL,
  primary key(id_universidad),
  foreign key (id_ciudad) references ciudad (id_ciudad)
  ON DELETE CASCADE ON UPDATE CASCADE
);
Create table usuario (
  email varchar (50) NOT NULL,
  nombre varchar (60) NOT NULL,
  passw varchar(50) NOT NULL,
  universidad_origen int NOT NULL,
  universidad_destino int NOT NULL,
  ciudad_destino int NOT NULL,
  Primary Key (email),
  foreign key (universidad_origen) references universidad (id_universidad)
  ON DELETE NO Action ON UPDATE No action,
  foreign key (universidad_destino) references universidad (id_universidad)
  ON DELETE NO Action ON UPDATE No action,
  foreign key (ciudad_destino) references ciudad (id_ciudad)
  ON DELETE NO Action ON UPDATE No action
);
Create table comentario(
  id_comentario int not null auto_increment,
  texto varchar(500),
  fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  email_usuario varchar(50) NOT NULL,
  id_ciudad int NOT NULL,
  tipo VARCHAR(50) NOT NULL CHECK (tipo IN ('cultura', 'comida', 'fiesta','transporte')),
  Primary Key (id_comentario),
  foreign key (email_usuario) references usuario (email)
  ON DELETE NO Action ON UPDATE No action,
  foreign key (id_ciudad) references ciudad (id_ciudad)
  ON DELETE NO Action ON UPDATE No action
);
Create table likes(
  email_usuario Varchar (50) NOT NULL,
  id_comentario int not null,
  primary key (email_usuario,id_comentario),
  foreign key (email_usuario) references usuario (email)
  ON DELETE NO Action ON UPDATE No action,
  foreign key (id_comentario) references comentario (id_comentario)
  ON DELETE NO Action ON UPDATE No action
);
Create table chat(
  id_chat int not null auto_increment,
  num_participantes int not null,
  primary key (id_chat)
);
```

```

Create table mensaje(
  id_mensaje int not null auto_increment,
  texto varchar (300),
  fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  id_chat int not null,
  email_usuario varchar(50) not null,
  primary key (id_mensaje),
  foreign key (id_chat) references chat (id_chat)
    ON DELETE CASCADE ON UPDATE CASCADE,
  foreign key (email_usuario) references usuario (email)
    ON DELETE NO Action ON UPDATE NO action
);

Create table pertenece(
  email_usuario varchar(50) NOT NULL,
  id_chat int not null,
  Primary Key (email_usuario,id_chat),
  foreign key (email_usuario) references usuario (email)
    ON DELETE CASCADE ON UPDATE CASCADE,
  foreign key (id_chat) references chat (id_chat)
    ON DELETE CASCADE ON UPDATE CASCADE
);

Create table Valoraciones(
  coste_vida int,
  transporte int,
  ambiente int,
  comida int,
  cultura int,
  clima int,
  tamaño int,
  email varchar(50) NOT NULL,
  id_ciudad int NOT NULL,
  primary key(id_ciudad,email),
  foreign key (email) references usuario (email)
    ON DELETE CASCADE ON UPDATE CASCADE,
  foreign key (id_ciudad) references ciudad (id_ciudad)
    ON DELETE CASCADE ON UPDATE CASCADE
);

Create table referente (
  id_chat INT NOT NULL,
  id_referencia_ciudad INT,
  id_referencia_universidad INT,
  tipo_referencia VARCHAR(50) CHECK (tipo_referencia IN ('ciudad', 'universidad')),
  tipo_universidad VARCHAR(50) CHECK (tipo_universidad IN ('origen', 'destino')),
  Primary key(id_chat),
  foreign key (id_chat) references chat (id_chat)
    ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (id_referencia_ciudad) REFERENCES ciudad (id_ciudad)
    ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (id_referencia_universidad) REFERENCES universidad (id_universidad)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT fk_referencia_tipo CHECK (
    (tipo_referencia = 'ciudad' AND id_referencia IN (SELECT id_ciudad FROM ciudad)) OR
    (tipo_referencia = 'universidad' AND id_referencia IN (SELECT id_universidad FROM
universidad))
  ),
  CONSTRAINT fk_notNull CHECK (
    (id_referencia_ciudad NOT LIKE NULL) OR (id_referencia_universidad NOT LIKE NULL)
  )
);

Create table procesos(
  id_proceso int NOT NULL auto_increment,
  titulo varchar (50) NOT NULL,
  descripcion varchar (500),
  fecha_vencimiento date,
  tipo VARCHAR(15) NOT NULL CHECK (tipo IN ('origen','destino')),
  id_universidad int NOT NULL,
  primary key (id_proceso),
  foreign key (id_universidad) references universidad (id_universidad)
  On delete cascade on update cascade
);

Create table completa(
  id_proceso int NOT NULL,
  email varchar(50) NOT NULL,
  primary key(id_proceso,email),
  foreign key(id_proceso) references procesos(id_proceso)
  ON DELETE cascade ON UPDATE cascade,
  foreign key(email) references usuario(email)
  ON DELETE cascade ON UPDATE cascade
);

```

Archivos Página WEB:

PantallaInicio.php Formato y style con CSS:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Página principal</title>
</head>
<style>
body {
font-family: Arial, sans-serif;
background-color: #fff8e1;
margin: 0;
padding: 0;
}
header{
background-color: #34495e;
height: 50px;
margin-bottom: 0; /* Margen inferior */
top: 0;
width: 100%;
}
h2{
color: black;
text-align: center;
font-family: Berlin Sans FB;
font-size: 50px;
background-color: white;
margin-left: 300px;
margin-right: 300px;
}
.menu {
list-style-type: none;
padding: 0;
margin: 0;
margin-top: 10px;
text-align: center;
}
.menu li {
display: inline-block; /* Para que los elementos estén en línea */
}
.menu li a {
display: block; /* Para que los enlaces ocupen todo el espacio del elemento <li> */
text-decoration: none; /* Quitar subrayado de los enlaces */
padding: 10px 20px; /* Ajustar el relleno para dar espacio alrededor del texto */
background-color: #4CAF50; /* Color de fondo */
color: white; /* Color del texto */
border: 1px solid #4CAF50; /* Borde del botón */
border-radius: 5px; /* Bordes redondeados */
margin-right: 10px; /* Espacio entre botones */
width: 150px;
}
.menu li a:hover {
background-color: #45a049; /* Cambiar color de fondo al pasar el ratón */
}
.inicio-sesion{
float: right;
list-style-type: none;
}
/* Estilos para el botón de inicio de sesión */
.inicio-sesion a {
background-color: #007bff;
border-color: #007bff;
}
.inicio-sesion a:hover {
background-color: #0056b3;
border-color: #0056b3;
}
/* Estilos para el botón de cerrar sesión */
.cerrar-sesion {
float: right;
list-style-type: none;
}
.cerrar-sesion a {
background-color: #dc3545;
border-color: #dc3545;
}
.cerrar-sesion a:hover {
background-color: #c82333;
border-color: #c82333;
}
.color{
color: #B6FDFE;
}
.marco {
border: none;
padding: 10px;
border-radius: 10px;
width: 95%;
}
.marco2{
border: 2px solid #ccc;
padding: 20px;
border-radius: 10px;
width: 85%;
margin: 10px auto;
position: center;
background-color: #4e342e;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Sombra suave */
text-align: center;
}
.marco3{
border: 2px solid #ccc;
padding: 20px;
border-radius: 10px;
width: 750px;
margin: 10px auto;
position: center;
background-color: #fff8e1;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Sombra suave */
text-align: center;
}
</style>
```

PantallaInicio.php Funcionalidad con html y php:

```
<body>
<header>
  <?php
  // Iniciar la sesión
  session_start();

  if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
  }
  $current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
  $_SESSION['anterior'] = $current_url;

  echo '<div class="color">';
  // Verificar si existe la variable de sesión 'usuario'
  if(isset($_SESSION['usuario'])) {
    // Recuperar el nombre de usuario de la sesión
    $nombre_usuario = $_SESSION['usuario'];
    // Imprimir el nombre de usuario
    echo "Bienvenido, $nombre_usuario"; ?>
    <li class="cerrar-sesion"><a href="cerrarsesion.php">Cerrar Sesión</a></li>
  } else {
    // Si no existe la variable de sesión 'usuario', el usuario no ha iniciado sesión
    echo "No has iniciado sesión."; ?>
    <li class="inicio-sesion"><a href="https://localhost/codigos/PantallaInicioSesion2.php">Iniciar
    Sesión</a></li>
  }
  <?php
  ?>
</div>
<h2>Bienvenido a Planerasmus</h2>
</header>

<div class="marco">
<ul class="menu">
  <div class="marco2">
    <div class="color">
      <?php
      echo "Acceda a foros de estudiantes para obtener información:"
      ?>
    </div>
    <div class="marco3">
    <li><a href="foros.php">Foros</a></li>
    </div>
  </div>
  <div class="marco2">
    <div class="color">
      <?php
      echo "Gestione sus procesos burocráticos:"
      ?>
    </div>
    <div class="marco3">
    <li><a href="acceso_procesos.php">Procesos</a></li>
    </div>
  </div>
  <div class="marco2">
    <div class="color">
      <?php
      echo "Acceda a sus chats:"
      ?>
    </div>
    <div class="marco3">
    <li><a href="acceso_chat.php">Chats</a></li>
    </div>
  </div>
  <div class="marco2">
    <div class="color">
      <?php
      echo "Acceda a las valoraciones realizadas por los estudiantes:"
      ?>
    </div>
    <div class="marco3">
    <li><a href="valoraciones.php">Valoraciones</a></li>
    </div>
  </div>
</ul>
</div>
</body>
</html>
```

PantallaInicioSesion.php, Estilo con CSS:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<title>Inicio de Sesión</title>
<style>
<body {
  font-family: Arial, sans-serif;
  background-color: #ecf0f1;
  margin: 0;
  padding: 0;
}
header{
  background-color: #34495e;
  height: 80px;
  margin-bottom: 0;
  top: 0;
  width: 100%;
}
.container {
  max-width: 400px;
  margin: 50px auto;
  padding: 20px;
  background-color: #fff;
  border-radius: 5px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h2 {
  text-align: center;
  color: #333;
}
form {
  text-align: center;
}
input[type="text"],
input[type="password"] {
  width: 100%;
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-sizing: border-box;
}
input[type="submit"] {
  background-color: #4CAF50;
  color: #fff;
  border: none;
  padding: 15px 20px;
  cursor: pointer;
  border-radius: 5px;
  font-size: 16px;
  transition: background-color 0.3s;
}
input[type="submit"]:hover {
  background-color: #45a049;
}
.register-btn {
  margin-top: 20px;
}
.register-btn a {
  display: inline-block;
  background-color: #007bff;
  color: #fff;
  text-decoration: none;
  padding: 15px 20px;
  border-radius: 5px;
  transition: background-color 0.3s;
}
.register-btn a:hover {
  background-color: #0056b3;
}
.button-container {
  position: fixed;
  top: 10px;
  right: 10px;
  z-index: 9999;
}
.button {
  padding: 10px 20px;
  margin-left: 10px;
  background-color: #5c9ded;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
.button-back {
  background-color: #e74c3c;
}
.titulo{
  top: 5px;
  font-weight: bold;
  font-size: 35px;
  text-align: center;
  color: #B6FDFD;
}
.usuario, .sesion {
  color: #FF0616;
  text-decoration: none;
  margin-left: 20px;
}
.usuario {
  font-size: 17px;
}
.color{
  color: #B6FDFD;
}
</style>
```

PantallaInicioSesion.php Funcionalidad con html y php:

```
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>

<?php
    session_start();

    if(isset($_SESSION['anterior'])) {
        $url_anterior = $_SESSION['anterior'];
    }

    if(isset($_GET['error'])) {
        // Recuperar el error
        $error = $_GET['error'];

        // Imprimir el error
        echo "<p style='color: red; text-align: center;'>ERROR: $error</p>";
    }
?>

<div class="titulo">
    <?php
        echo "PLANERASMUS";
    ?>
</div>
</header>
<div class="container">
    <h2>Inicio de Sesión</h2>
    <form method="post" action="inicio_sesion.php">
        <label for="email">E-mail:</label><br>
        <input type="text" id="email" name="email" required><br><br>
        <label for="password">Contraseña:</label><br>
        <input type="password" id="Contraseña" name="Contraseña" required><br><br>
        <input type="submit" name="InicioSesion" value="Iniciar sesión"><br><br>
    </form>

    <div class="register-btn">
        <a href="RegistroUsuario2.php">Registrarse</a>
    </div>
</div>
<div class="button-container">
    <!-- Botón "Atrás" -->
    <a button onclick="volverAtras()" class="button button-back">Atras</button>
</a><!-- Botón "Home" -->
    <a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>
```

Iniciar_Sesion.php Funcionalidad backend, en html y php:

```

<?php
session_start();

if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

$email = $password = "";
$errores = array();
$dadosOK = true;

$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["email"])) {
        $error_message = "El campo email es requerido";
        header("Location: https://localhost/codigos/PantallaInicioSesion.php?
error=".urlencode($error_message));
        exit();
    }else {
        $email = test_input($_POST["email"]);
        $seguro = $conn->prepare("SELECT passw FROM usuario where email = ?");
        $seguro->bind_param("s", $email);
        $seguro->execute();
        $result = $seguro->get_result();
        $num_filas = $result->num_rows;
        if ($num_filas == 1){
            if (empty($_POST["Contraseña"])) {
                $error_message = "El campo contraseña es requerido";
                header("Location: https://localhost/codigos/PantallaInicioSesion.php?
error=".urlencode($error_message));
                exit();
            }else{
                $fila = $result->fetch_assoc();
                $contra = $fila['passw'];
                $password = test_input($_POST["Contraseña"]);
                if($password == $contra){
                    $_SESSION['usuario'] = $email;
                    header("Location: $url_anterior");
                    exit();
                }
                else{
                    $error_message = "Contraseña incorrecta";
                    header("Location: https://localhost/codigos/PantallaInicioSesion2.php?
error=".urlencode($error_message));
                    exit();
                }
            }
        }
        else{
            $error_message = "No existe ningún usuario registrado con ese email";
            header("Location: https://localhost/codigos/PantallaInicioSesion2.php?
error=".urlencode($error_message));
            exit();
        }
    }
}
?>

```

Cerrarsesion.php Funcionalidad con html y php:

```
<?php
session_start();
$_SESSION['usuario'] = null;
$error_message = "Sesión cerrada";
header("Location: https://localhost/codigos/PantallaInicio.php")
?>
```

RegistroUsuario.php Estilo con CSS:

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Registro en la Aplicación</title>
<style>
body {
font-family: Arial, sans-serif;
background-color: #ecf0f1;
margin: 0;
padding: 0;
}
header{
background-color: #34495e;
height: 50px;
margin-bottom: 0;
top: 0;
width: 100%;
}
.container {
max-width: 600px;
margin: 50px auto;
padding: 20px;
background-color: #fff;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h2 {
text-align: center;
color: #333;
}
form {
text-align: center;
}
input[type="text"],
input[type="password"] {
width: 100%;
padding: 10px;
margin: 10px 0;
border: 1px solid #ccc;
border-radius: 5px;
box-sizing: border-box;
}

input[type="submit"] {
background-color: #007bff;
color: #fff;
border: none;
padding: 15px 20px;
cursor: pointer;
border-radius: 5px;
font-size: 16px;
transition: background-color 0.3s;
}
input[type="submit"]:hover {
background-color: #0056b3;
}
select {
width: 100%;
padding: 10px;
margin: 10px 0;
border: 1px solid #ccc;
border-radius: 5px;
box-sizing: border-box;
}
.button-container {
position: fixed;
top: 10px;
right: 10px;
z-index: 9999;
}
.button {
padding: 10px 20px;
margin-left: 10px;
background-color: #5c9ded;
color: #fff;
border: none;
border-radius: 5px;
cursor: pointer;
}
.button-back {
background-color: #74c3c;
}
.titulo{
top: 5px;
font-weight: bold;
font-size: 35px;
text-align: center;
color: #B6FDFF;
}
</style>

```

RegistroUsuario.php Funcionalidad en html y php:

```

</body>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>
<div class="titulo">
    <?php
    echo "PLANERASMUS";
    ?>
</div>
</header>
<div class="container">
    <h2>Registro en la Aplicación</h2>

    <?php
    session_start();
    if(isset($_SESSION['anterior'])) {
        $url_anterior = $_SESSION['anterior'];
    }
    $current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
    $_SESSION['anterior'] = $current_url;

    if(isset($_GET['error'])) {
        // Recuperar el error
        $error = $_GET['error'];

        // Imprimir el error
        echo "<p style='color: red; text-align: center;'>Error: $error</p>";
    }

    // Conexión a la base de datos
    $mysqli = mysqli_connect("localhost:3306", "Plan
    Erasmus", "ErasmusBudapest", "planerasmus");
    $mysqli->set_charset("utf8");
    $resultado = mysqli_query($mysqli, "select * from universidad order by nombre");
    ?>

    <form method="post" action="input_datos.php">
    <label for="name">Name:</label><br>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">E-mail:</label><br>
    <input type="text" id="email" name="email" required><br><br>

    <label for="password">Contraseña:</label><br>
    <input type="password" id="Contraseña" name="Contraseña" required><br><br>

    <label for="UniversidadOrigen">Universidad Origen:</label><br>
    <select id="UniversidadOrigen" name="UniversidadOrigen" required>
    <?php
    while($fila = $resultado->fetch_assoc()){
        $id = $fila['id_universidad'];
        $nombre = $fila['nombre'];
        echo "<option value='$id'$>$nombre</option>";
    }
    ?>
    </select><br><br>

    <label for="UniversidadDestino">Universidad Destino:</label><br>
    <select id="UniversidadDestino" name="UniversidadDestino" required>
    <?php
    $resultado = mysqli_query($mysqli, "select * from universidad order by nombre");
    while($fila = $resultado->fetch_assoc()){
        $id = $fila['id_universidad'];
        $nombre = $fila['nombre'];
        echo "<option value='$id'$>$nombre</option>";
    }
    ?>
    </select><br><br>

    <input type="submit" name="Registro" value="Registrarse"><br><br>
    </form>
</div>
<div class="button-container">
    <!-- Botón "Atrás" -->
    <a button onclick="volverAtras()" class="button button-back">Atrás</button></a>
    <!-- Botón "Home" -->
    <a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>

```

Input_datos.php Funcionalidad Backend con html y php:

```

<?php
session_start();

if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

// define variables and set to empty values
$name = $email = $password = $ciudad = $universidadO = $universidadD = "";
$errores = array();
$datosOK = true;

$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $errores[] = "El campo nombre es requerido";
        $datosOK = false;
    } else {
        $name = test_input($_POST["name"]);
    }

    if (empty($_POST["email"])) {
        $errores[] = "El campo email es requerido";
        $datosOK = false;
    } else {
        $email = test_input($_POST["email"]);
        $seguro = $conn->prepare("SELECT * FROM usuario where email = ?");
        $seguro->bind_param("s", $email);
        $seguro->execute();
        $result = $seguro->get_result();
        $num_filas = $result->num_rows;
        if($num_filas != 0) {
            $errores[] = "Ya hay un usuario registrado con ese email";
            $datosOK = false;
        }
    }

    if (empty($_POST["Contraseña"])) {
        $errores[] = "El campo contraseña es requerido";
        $datosOK = false;
    } else {
        $password = test_input($_POST["Contraseña"]);
    }

    if (empty($_POST["UniversidadOrigen"])) {
        $errores[] = "El campo universidad origen es requerido";
        $datosOK = false;
    } else {
        $universidadO = test_input($_POST["UniversidadOrigen"]);
    }

    if (empty($_POST["UniversidadDestino"])) {
        $errores[] = "El campo universidad destino es requerido";
        $datosOK = false;
    } else {
        $universidadD = test_input($_POST["UniversidadDestino"]);
    }
}

if($datosOK == true){
    $seguro = $conn->prepare("SELECT id_ciudad FROM universidad where id_universidad = $universidadD");
    $seguro->execute();
    $result = $seguro->get_result();
    $fila = $result->fetch_assoc();
    $ciudad = $fila['id_ciudad'];

    $seguro = $conn->prepare("INSERT INTO usuario (email, nombre, ciudad_destino,
universidad_origen, universidad_destino, passw)
VALUES (?, ?, ?, ?, ?)");
    $seguro->bind_param("sslls", $email, $name, $ciudad, $universidadO, $universidadD,
$password);
    $result = $seguro->execute();
    $seguro->close();
    $_SESSION['usuario'] = $email;
    //RECUPERAMOS LN SUS CHATS
    //Recuperamos el chat de la ciudad destino
    $seguro = $conn->prepare("Select id_chat from referente where id_referencia_ciudad =
? and tipo_referencia = 'ciudad'");
    $seguro->bind_param("l", $ciudad);
    $seguro->execute();
    $result = $seguro->get_result();
    if($result->num_rows==1){
        $fila = $result->fetch_assoc();
        $chat = $fila['id_chat'];
        //Agregamos al nuevo usuario al chat
        $seguro = $conn->prepare("INSERT INTO pertenece (email_usuario, id_chat) VALUES
(?, ?)");
        $seguro->bind_param("sl", $email, $chat);
        $result = $seguro->execute();
        $seguro = $conn->prepare("UPDATE chat SET num_participantes = num_participantes +
1 where id_chat = ?");
        $seguro->bind_param("l", $chat);
        $result = $seguro->execute();
        $seguro->close();
    }
    else{
        echo "Error al recuperar el id del chat de ciudad";
    }
    $seguro = $conn->prepare("Select id_chat from referente where
id_referencia_universidad = ? and tipo_referencia = 'universidad' and tipo_universidad =
'origen'");
    $seguro->bind_param("l", $universidadO);
    $seguro->execute();
    $result = $seguro->get_result();
    if($result->num_rows==1){
        $fila = $result->fetch_assoc();
        $chat = $fila['id_chat'];
        //Agregamos al nuevo usuario al chat
        $seguro = $conn->prepare("INSERT INTO pertenece (email_usuario, id_chat) VALUES
(?, ?)");
        $seguro->bind_param("sl", $email, $chat);
        $result = $seguro->execute();
        $seguro = $conn->prepare("UPDATE chat SET num_participantes = num_participantes +
1 where id_chat = ?");
        $seguro->bind_param("l", $chat);
        $result = $seguro->execute();
        $seguro->close();
    }
    else{
        echo "Error al recuperar el id del chat de la universidad origen";
    }
    $seguro = $conn->prepare("Select id_chat from referente where
id_referencia_universidad = ? and tipo_referencia = 'universidad' and tipo_universidad =
'destino'");
    $seguro->bind_param("l", $universidadD);
    $seguro->execute();
    $result = $seguro->get_result();
    if($result->num_rows==1){
        $fila = $result->fetch_assoc();
        $chat = $fila['id_chat'];
        //Agregamos al nuevo usuario al chat
        $seguro = $conn->prepare("INSERT INTO pertenece (email_usuario, id_chat) VALUES
(?, ?)");
        $seguro->bind_param("sl", $email, $chat);
        $result = $seguro->execute();
        $seguro = $conn->prepare("UPDATE chat SET num_participantes = num_participantes +
1 where id_chat = ?");
        $seguro->bind_param("l", $chat);
        $result = $seguro->execute();
        $seguro->close();
    }
    else{
        echo "Error al recuperar el id del chat de la universidad destino";
    }

    echo "Se ha registrado correctamente";
    header("Location: https://localhost/codigos/PantallaInicio.php");
    exit();
} else {
    echo "Error al registrar usuario en la BBDD";
    header("Location: https://localhost/codigos/RegistroUsuario2.php");
    exit();
}
}
else{
    $error_message = implode("<br>", $errores);
    header("Location: http://localhost/codigos/RegistroUsuario2.php
error=" . urlencode($error_message));
    exit();
}
$conn->close();
}

```

Acceso_procesos.php Estilo con CSS:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Acceso a los chats</title>
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  .boton {
    display: inline-block;
    padding: 10px 20px;
    margin: 10px;
    font-size: 18px;
    text-align: center;
    text-decoration: none;
    border: 2px solid #4CAF50;
    color: #4CAF50;
    border-radius: 8px;
    transition: background-color 0.3s;
    background-color: white;
  }
  .boton:hover {
    background-color: #4CAF50;
    color: white;
  }
  .marco {
    border: 2px solid #ccc;
    padding: 20px;
    border-radius: 10px;
    width: fit-content;
    margin: 0 auto;
    background-color: rgba(255, 255, 255, 0.9);
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  }
  body {
    background-color: #f2f2f2;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #B6FDFD;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  .color{
    color: #B6FDFD;
  }
</style>
```

Acceso_procesos.php Funcionalidad con Html y PHP:

```
<body>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>
<?php
// Iniciar la sesión
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;

?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo '<button onclick="location.href=\'cerrarsesion.php\'" class="sesion">Cerrar
Sesión</button>';
} else {
    echo "No has iniciado sesión.";
    echo '<button
onclick="location.href=\'https://localhost/codigos/PantallaInicioSesion2.php\'"
class="sesion">Iniciar Sesión</button>';
}
?>
</div>
<div class="titulo">
<?php
    echo "PLANERASMUS";
?>
</div>
</header>
<h2 style="text-align: center;"> Bienvenido a la gestión de procesos</h2>
<div class="marco">
    <a href="devolverProcesos.php?tipo=origen" class="boton">Procesos de universidad
origen</a>;
    <a href="devolverProcesos.php?tipo=destino" class="boton">Procesos de universidad
destino</a>;
</div>

<div class="button-container">
    <!-- Botón "Atrás" -->
    <a button onclick="volverAtras()" class="button button-back">Atras</button></a>
    <!-- Botón "Home" -->
    <a href="PantallaInicio.php" class="button">Home</a>
</div>

</body>
```

DevolverProceso.php Estilo con CSS:

```

<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  .proceso {
    border: 1px solid #ccc;
    padding: 10px;
    margin-top: 10px;
    margin-left: 100px;
    margin-right: 100px;
    text-align: center;
    background-color: #FFFFFF8;
  }
  .proceso h3 {
    font-weight: bold;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #B6FDFD;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  .color{
    color: #B6FDFD;
  }
</style>

```

DevolverProcesos.php Funcionalidad con HTML y PHP:

```

<body>
<header>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;
?>
<div class="color">
<?php
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo '<button onclick="location.href=\'cerrarsesion.php\'" class="sesion">Cerrar
Sesión</button>';
} else {
    echo "No has iniciado sesión.";
    echo '<button
onclick="location.href=\'https://localhost/codigos/PantallaInicioSesion2.php\'"
class="sesion">Iniciar Sesión</button>';
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
</header>
<?php
$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");
if (isset($_GET['tipo'])){
    $tipo = test_input($_GET['tipo']);
    if($tipo == "origen"){
        $seguro = $conn->prepare("SELECT p.*
usuario s
from procesos p inner join universidad u inner join
s.universidad_origen = u.id_universidad
on p.id_universidad = u.id_universidad and
where p.tipo = ? and s.email= ?
order by fecha_vencimiento;");
    }
    else{
        $seguro = $conn->prepare("SELECT p.*
usuario s
from procesos p inner join universidad u inner join
s.universidad_destino = u.id_universidad
on p.id_universidad = u.id_universidad and
where p.tipo = ? and s.email= ?
order by fecha_vencimiento;");
    }
    $seguro->bind_param("ss",$tipo,$nombre_usuario);
    $seguro->execute();
    $result = $seguro->get_result();
    while($fila = $result->fetch_assoc()){
        $id_proceso = $fila['id_proceso'];
        $titulo = $fila['titulo'];
        $fecha = $fila['fecha_vencimiento'];
        $descripcion = $fila['descripcion'];
        echo "<div class='proceso'>";
        echo "<h3>$titulo</h3>";
        echo "<p>$descripcion</p>";
        echo "<p>Fecha vencimiento: $fecha</p>";
    }
    $seguro2 = $conn->prepare("Select * from completa where email = ? and
id_proceso = ?;");
    $seguro2->bind_param("si", $nombre_usuario, $id_proceso);
    $seguro2->execute();
    $result2 = $seguro2->get_result();
    if($result2->num_rows != 1){
        ?>
        <form action="marcar_proceso.php" method="post">
        <input type="hidden" name="proceso" value="<?php echo
$id_proceso; ?>">
        <button type="submit">
        <?php
        echo "Marcar como completo";
        ?>
        </button>
        </form>
        <?php
        }
        else{
            echo "Completado";
        }
        echo "</div>";
    }
}
?>
<div class="button-container">
<!-- Botón "Atrás" -->
<a button onclick="volverAtras()" class="button button-back">Atras</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>

```

Marcar_proceso.php Funcionalidad Backend con PHP:

```
<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

// Iniciar la sesión
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
}
else{
    header("Location: $url_anterior");
    exit();
}
$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";
$ciudad = "";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $proceso = test_input($_POST["proceso"]);
    $seguro = $conn->prepare("Insert into completa (email,id_proceso) VALUES (?,?);");
    $seguro->bind_param("si", $nombre_usuario, $proceso);
    $result = $seguro->execute();

    $seguro->close();
    $conn->close();
    header("Location: $url_anterior");
    exit();
}
?>
```

Foros.php Estilo con CSS:

```
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  .container {
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #B6FDFD;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  h2 {
    text-align: center;
    color: #333;
  }
  .mensaje {
    text-align: center;
    margin-bottom: 20px;
  }
  form {
    text-align: center;
  }
  select {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    width: 100%;
    max-width: 300px;
  }
  input[type="submit"] {
    padding: 10px 20px;
    background-color: #4CAF50;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
  }
  input[type="submit"]:hover {
    background-color: #45a049;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .color{
    color: #B6FDFD;
  }
  .sesion{
    color: #FF0616;
  }
</style>
```

Foros.php Funcionalidad con HTML y PHP:

```
<body>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>

<?php
// Iniciar la sesión
session_start();

if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;
?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo '<button onclick="location.href=\'cerrarsesion.php\'" class="sesion">Cerrar
Sesión</button>';
} else {
    echo "No has iniciado sesión.";
    echo '<button
onclick="location.href=\'https://localhost/codigos/PantallaInicioSesion2.php\'"
class="sesion">Iniciar Sesión</button>';
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
<?php

$i = 10;
$_SESSION['num_coments'] = $i;
?>
</header>
<div class="container">
<h2>Bienvenido a los foros</h2>
<form method="post" action="acceso_foros.php">
Seleccione la ciudad de la cuál quiere ver comentarios:
<select name="Ciudad" required>
<?php
    $mysqli = mysqli_connect("localhost:3306","Plan
Erasmus","ErasmusBudapest","planerasmus");
    $mysqli->set_charset("utf8");
    $resultado = mysqli_query($mysqli, "select * from ciudad order by nombre");

    while($fila = $resultado->fetch_assoc()){
        $id = $fila['id_ciudad'];
        $nombre = $fila['nombre'];
        $pais = $fila['pais'];
        echo "<option value='$id'$>$nombre,$pais</option>";
    }
?>
</select><br><br>
<p><input type="submit" name="Acceso" value="Acceder a foro"></p>
</form>
</div>
<div class="button-container">
<!-- Botón "Atrás" -->
<a button onclick="volverAtras()" class="button button-back">Atrás</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>
```

Acceso_foros.php Estilo con CSS:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
body {
  font-family: Arial, sans-serif;
  background-color: #ecf0f1;
  margin: 0;
  padding: 0;
}
header{
  background-color: #34495e;
  height: 80px;
  margin-bottom: 0;
  top: 0;
  width: 100%;
}
.menu {
  list-style-type: none;
  padding: 0;
  margin: 20px 0;
  text-align: center;
}
.menu li {
  display: inline;
}
.menu li a {
  display: inline-block;
  padding: 10px 20px;
  background-color: #007bff;
  color: #fff;
  text-decoration: none;
  border-radius: 5px;
  margin: 0 5px;
}
.menu li a:hover {
  background-color: #0056b3;
}
.button-container {
  position: fixed;
  top: 10px;
  right: 10px;
  z-index: 9999;
}
.button {
  padding: 10px 20px;
  margin-left: 10px;
  background-color: #5c9ded;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
.button-back {
  background-color: #e74c3c;
}
.marco {
  border: 2px solid #ccc;
  padding: 20px;
  border-radius: 10px;
  width: fit-content;
  margin: 85px auto;
  position: center;
  background-color: rgba(255, 255, 255, 0.9);
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  text-align: left;
}
.marco2{
  border: 2px solid #ccc;
  padding: 20px;
  border-radius: 10px;
  width: 800px;
  margin: 10px auto;
  position: center;
  background-color: #a3c4dc;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  text-align: center;
}
.texto{
  font-weight: bold;
  font-size: 20px;
  text-align: center;
  margin-bottom: 50px;
}
.color{
  color: #B6FDFD;
}
.usuario, .sesion {
  color: #FF0616;
  text-decoration: none;
  margin-left: 20px;
}
.usuario {
  font-size: 17px;
}
.titulo{
  top: 5px;
  font-weight: bold;
  font-size: 35px;
  text-align: center;
  color: #B6FDFD;
}
</style>
```

Acceso_foros.php Funcionalidad con HTML y PHP:

```
<body>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>
<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
session_start();

if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;
?>
<div class="color">
<?php
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo '<button onclick="location.href=\'cerrarsesion.php\'" class="sesion">Cerrar
Sesión</button>';
} else {
    echo "No has iniciado sesión.";
    echo '<button
onclick="location.href=\'https://localhost/codigos/PantallaInicioSesion2.php\'"
class="sesion">Iniciar Sesión</button>';
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $ciudad = test_input($_POST["Ciudad"]);
    $_SESSION['ciudad'] = $ciudad;
}
else{
    $ciudad=$_SESSION['ciudad'];
}
?>
</header>
<div class="marco">
<div class="texto">
<?php
echo "Qué tipo de información desea consultar?";
?>
</div>
<ul class="menu">
<div class="marco2">
<?php
echo "Consulte información sobre la comida:";
?>
<li><a href="devolverComentarioF.php?tipo=comida">Comida</a></li>
</div>
<div class="marco2">
<?php
echo "Consulte información sobre la cultura, monumentos o puntos de interés:";
?>
<li><a href="devolverComentarioF.php?tipo=cultura">Cultura</a></li>
</div>
<div class="marco2">
<?php
echo "Consulte información sobre lugares de ocio, bares o discotecas:";
?>
<li><a href="devolverComentarioF.php?tipo=fiesta">Fiesta</a></li>
</div>
<div class="marco2">
<?php
echo "Consulte información sobre cómo funciona el transporte en la ciudad:";
?>
<li><a href="devolverComentarioF.php?tipo=transporte">Transporte</a></li>
</div>
</ul>
</div>
<div class="button-container">
<!-- Botón "Atrás" -->
<a button onclick="volverAtras()" class="button button-back">Atras</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>
```

DevolverComentario.php Estilo con CSS:

```
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  h2 {
    text-align: center;
    color: #333;
  }
  .comment {
    margin-bottom: 20px;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #f9f9f9;
  }
  .comment .info {
    font-size: 14px;
    color: #666;
    margin-bottom: 5px;
  }
  .comment .text {
    font-size: 16px;
    line-height: 1.5;
  }
  input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
  }
  input[type="submit"]:hover {
    background-color: #45a049;
  }
  .like-button {
    float: right;
    margin-top: 10px;
  }
  .like-button button {
    background-color: #007bff;
    color: white;
    border: none;
    padding: 5px 10px;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
  }
  .like-button button:hover {
    background-color: #0056b3;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .button-wrapper {
    text-align: center;
    margin-top: 20px;
  }
  .button-wrapper form {
    display: inline-block;
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #86F0FF;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  .color{
    color: #86F0FF;
  }
</style>
```


Dar_like.php Funcionalidad Backend con PHP:

```
<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

// Iniciar la sesión
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
}
else{
    header("Location: $url_anterior");
    exit();
}
$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";
$ciudad = "";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $comentario = test_input($_POST["comentario_id"]);
    $seguro = $conn->prepare("Select * from likes where email_usuario = ? and id_comentario = ?;");
    $seguro->bind_param("si", $nombre_usuario, $comentario);
    $seguro->execute();
    $result = $seguro->get_result();
    if($result->num_rows == 1){
        $seguro = $conn->prepare("Delete from likes where email_usuario = ? and id_comentario = ?");
        $seguro->bind_param("si", $nombre_usuario, $comentario);
        $result = $seguro->execute();
    }
    else{
        $seguro = $conn->prepare("Insert into likes (email_usuario,id_comentario) VALUES (?,?,?);");
        $seguro->bind_param("si", $nombre_usuario, $comentario);
        $result = $seguro->execute();
    }
    $seguro->close();
    $conn->close();
    header("Location: $url_anterior");
    exit();
}
?>
```

Añadir_comentario.php Estilo con CSS:

```

<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
body {
font-family: Arial, sans-serif;
background-color: #ecf0f1;
margin: 0;
padding: 0;
}
header{
background-color: #34495e;
height: 80px;
margin-bottom: 0;
top: 0;
width: 100%;
}
h2 {
color: #333;
text-align: center;
margin-bottom: 30px;
}
form {
max-width: 400px;
margin: 0 auto;
background-color: #fff;
padding: 20px;
border-radius: 10px;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}
input[type="text"],
input[type="submit"] {
width: 100%;
padding: 10px;
margin-bottom: 20px;
border: 1px solid #ccc;
border-radius: 5px;
box-sizing: border-box;
}
input[type="submit"] {
background-color: #4CAF50;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s;
}
input[type="submit"]:hover {
background-color: #45a049;
}
.message {
text-align: center;
margin-top: 20px;
color: #007bff;
}
.button-container {
position: fixed;
top: 10px;
right: 10px;
z-index: 9999;
}
.button {
padding: 10px 20px;
margin-left: 10px;
background-color: #5c9ded;
color: #fff;
border: none;
border-radius: 5px;
cursor: pointer;
}
.button-back {
background-color: #e74c3c;
}
.titulo{
top: 5px;
font-weight: bold;
font-size: 35px;
text-align: center;
color: #B6FDFF;
}
.usuario, .sesion {
color: #FF0610;
text-decoration: none;
margin-left: 20px;
}
.usuario {
font-size: 17px;
}
.color{
color: #B6FDFF;
}
</style>

```

Añadir_comentario.php Funcionalidad con HTML y PHP:

```

<body>
<header>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>

<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;

if(isset($_SESSION['ciudad'])) {
    $ciudad = $_SESSION['ciudad'];
}

?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo '<button onclick="location.href=\'cerrarsesion.php\'" class="sesion">Cerrar
Sesión</button>';
} else {
    echo "No has iniciado sesión.";
    echo '<button
onclick="location.href=\'https://localhost/codigos/PantallaInicioSesion2.php\'"
class="sesion">Iniciar Sesión</button>';
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
</header>
<h2>Añada un comentario</h2>
<form method="post" action="añadir_comentarioBE.php?tipo=<?php echo $tipo?>">
    <label for="texto">¿Qué desea comentar?</label><br>
    <input type="text" id="texto" name="texto" required><br><br>
    <input type="submit" name="Coment" value="Comentar">
</form>
<div class="button-container">
    <!-- Botón "Atrás" -->
    <a button onclick="volverAtras()" class="button button-back">Atras</button></a>
    <!-- Botón "Home" -->
    <a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>

```

Añadir_comentarioBE.php Funcionalidad Backend:

```

<?php
session_start();

if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $texto = test_input($_POST["texto"]);
    if(isset($_SESSION['ciudad'])) {
        $ciudad = $_SESSION['ciudad'];
    }
    if(isset($_GET['tipo'])){
        $tipo = test_input($_GET["tipo"]);
    }
    else{
        echo "Error con el tipo";
        header("refresh:3; url=https://localhost/codigos/foros.php");
        exit();
    }
    if(isset($_SESSION['usuario'])) {
        // Recuperar el nombre de usuario de la sesión
        $nombre_usuario = $_SESSION['usuario'];
        $seguro = $conn->prepare("INSERT INTO comentario (texto,email_usuario,id_ciudad,tipo)
VALUES (?, ?, ?, ?);");
        $seguro->bind_param("ssis", $texto,$nombre_usuario,$ciudad,$tipo);
        $result = $seguro->execute();
        $id = $conn->insert_id;
        $seguro = $conn->prepare("INSERT INTO likes (email_usuario,id_comentario) VALUES
(?, ?);");
        $seguro->bind_param("si", $nombre_usuario,$id);
        $result = $seguro->execute();
        $seguro->close();
        $conn->close();
        if($result == TRUE){
            echo "Comentario realizado con éxito";
            header("Location: https://localhost/codigos/devolverComentarioF.php");
            exit();
        }
        else{
            echo "Ha surgido algún problema al añadir comentario";
            header("Location: https://localhost/codigos/foros.php");
            exit();
        }
    }
    else{
        echo "Por favor inicie sesión y vuelva a intartarlo";
        header("Location: https://localhost/codigos/PantallaInicioSesion2.php");
        exit();
    }
}

?>

```

Valoraciones.php Estilo con CSS:

```

<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  .container {
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  h2 {
    text-align: center;
    color: #333;
  }
  .mensaje {
    text-align: center;
    margin-bottom: 20px;
  }
  form {
    text-align: center;
  }
  select {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    width: 100%;
    max-width: 300px;
  }
  input[type="submit"] {
    padding: 10px 20px;
    background-color: #4CAF50;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
  }
  input[type="submit"]:hover {
    background-color: #45a049;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #B6F0FF;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  .color{
    color: #B6F0FF;
  }
</style>

```

Valoraciones.php Funcionalidad con HTML y PHP:

```

<body>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>
<div class="titulo">
    <?php
    echo "PLANERASMUS";
    ?>
</div>
</header>
<div class="container">
    <h2>Registro en La Aplicación</h2>

    <?php
    session_start();
    if(isset($_SESSION['anterior'])) {
        $url_anterior = $_SESSION['anterior'];
    }
    $current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
    $_SESSION['anterior'] = $current_url;

    if(isset($_GET['error'])) {
        // Recuperar el error
        $error = $_GET['error'];

        // Imprimir el error
        echo "<p style='color: red; text-align: center;'>Error: $error</p>";
    }

    // Conexión a la base de datos
    $mysqli = mysqli_connect("localhost:3306", "Plan
Erasmus", "ErasmusBudapest", "planerasmus");
    $mysqli->set_charset("utf8");
    $resultado = mysqli_query($mysqli, "select * from universidad order by nombre");
    ?>

<form method="post" action="input_datos.php">
    <label for="name">Name:</label><br>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">E-mail:</label><br>
    <input type="text" id="email" name="email" required><br><br>

    <label for="password">Contraseña:</label><br>
    <input type="password" id="Contraseña" name="Contraseña" required><br><br>

    <label for="UniversidadOrigen">Universidad Origen:</label><br>
    <select id="UniversidadOrigen" name="UniversidadOrigen" required>
    <?php
    while($fila = $resultado->fetch_assoc()){
        $id = $fila['id_universidad'];
        $nombre = $fila['nombre'];
        echo "<option value='$id'>$nombre</option>";
    }
    ?>
</select><br><br>

    <label for="UniversidadDestino">Universidad Destino:</label><br>
    <select id="UniversidadDestino" name="UniversidadDestino" required>
    <?php
    $resultado = mysqli_query($mysqli, "select * from universidad order by nombre");
    while($fila = $resultado->fetch_assoc()){
        $id = $fila['id_universidad'];
        $nombre = $fila['nombre'];
        echo "<option value='$id'>$nombre</option>";
    }
    ?>
</select><br><br>

    <input type="submit" name="Registro" value="Registrarse"><br><br>
</form>
</div>
<div class="button-container">
    <!-- Botón "Atrás" -->
    <a button onclick="volverAtras()" class="button button-back">Atras</button></a>
    <!-- Botón "Home" -->
    <a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>

```

DevolverValoracionesF.php Estilo con CSS:

```
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  h2 {
    color: #007bff;
  }
  .container {
    text-align: center;
    max-width: 600px;
    margin: 50 auto;
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  }
  .score {
    text-align: center;
    margin-bottom: 10px;
  }
  .btn {
    text-align: center;
    background-color: #007bff;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
  }
  .btn:hover {
    background-color: #0056b3;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #B6FDFD;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  .color{
    color: #B6FDFD;
  }
</style>
```

DevolverValoracionesF.php Funcionalidad con HTML y PHP:

```
</body>
<headers>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>

<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
session_start();

if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;
?>
<div class="color">
<?php
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo "<button onClick='location.href=\"cerrarsesion.php\"' class='sesion'>Cerrar Sesión</button>";
} else {
    echo "No has iniciado sesión.";
    echo "<button
onClick='location.href=\"https://localhost/codigos/PantallaInicioSesion2.php\"'
class='sesion'>Iniciar Sesión</button>";
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
</header>
<div class="container">
<?php

$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "Planerasmus";
$ciudad = "";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");

if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $ciudad = test_input($_POST["Ciudad"]);
    $_SESSION['ciudad'] = $ciudad;
}
else if(isset($_SESSION['ciudad'])) {
    // Recuperar el error
    $ciudad = $_SESSION['ciudad'];
    echo "Agregada la valoración";
}

$seguro = $conn->prepare("Select nombre from ciudad where id_ciudad = ?");
$seguro->bind_param("i", $ciudad);
$seguro->execute();
$result = $seguro->get_result();
$fila = $result->fetch_assoc();
$nombreCiudad = $fila['nombre'];
echo "<h2>Valoraciones de $nombreCiudad</h2>";
$seguro = $conn->prepare("Select avg(coste_vida), avg(transporte), avg(ambiente),
avg(comida),
    avg(cultura), avg(clima), avg(dimensiones)
    from valoraciones where id_ciudad = ?;");
$seguro->bind_param("i", $ciudad);
$seguro->execute();
$result = $seguro->get_result();
$fila = $result->fetch_assoc();
$coste = $fila['avg(coste_vida)'];
$costeR = round($coste,2);
$transporte = $fila['avg(transporte)'];
$transporteR = round($transporte,2);
$ambiente = $fila['avg(ambiente)'];
$ambienteR = round($ambiente,2);
$comida = $fila['avg(comida)'];
$comidaR = round($comida,2);
$cultura = $fila['avg(cultura)'];
$culturaR = round($cultura,2);
$clima = $fila['avg(clima)'];
$climaR = round($clima,2);
$dimensiones = $fila['avg(dimensiones)'];
$dimensionesR = round($dimensiones,2);

?>
<div class="score"><?php echo "Puntuación Coste de vida: ".$costeR."/10"?></div>
<div class="score"><?php echo "Puntuación Transporte: ".$transporteR."/10"?></div>
<div class="score"><?php echo "Puntuación ambiente: ".$ambienteR."/10"?></div>
<div class="score"><?php echo "Puntuación comida: ".$comidaR."/10"?></div>
<div class="score"><?php echo "Puntuación cultura: ".$culturaR."/10"?></div>
<div class="score"><?php echo "Puntuación clima: ".$climaR."/10"?></div>
<div class="score"><?php echo "Puntuación Tamaño: ".$dimensionesR."/10"?></div>

<form method="post" action="añadir_valoraciones.php">
    <input type="submit" class="btn" name="Añadir Valoraciones" value="Haga sus
valoraciones"></p>
</form>
</div>
<div class="button-container">
<!-- Botón "Atrás" -->
<a button onClick="volverAtras()" class="button button-back">Atras</button></a>

<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>
```

Añadir_valoraciones.php Estilo con CSS:

```
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  h2 {
    color: #007bff;
    text-align: center;
  }
  form {
    max-width: 500px;
    margin: auto;
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  }
  label {
    font-weight: bold;
  }
  select, input[type="submit"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-sizing: border-box;
  }
  input[type="submit"] {
    background-color: #007bff;
    color: #fff;
    border: none;
    cursor: pointer;
  }
  input[type="submit"]:hover {
    background-color: #0056b3;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #B6FDFD;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  .color{
    color: #B6FDFD;
  }
</style>
```

Añadir_valoraciones.php Funcionalidad con HTML y PHP:

```

<body>
<header>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>

<?php
    session_start();
    if(isset($_SESSION['anterior'])) {
        $url_anterior = $_SESSION['anterior'];
    }
    $current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
    $_SESSION['anterior'] = $current_url;

    if(isset($_SESSION['ciudad'])) {
        $ciudad = $_SESSION['ciudad'];
    }
    ?>
<div class="color">
<?php
    // Verificar si existe la variable de sesión 'usuario'
    if(isset($_SESSION['usuario'])) {
        $nombre_usuario = $_SESSION['usuario'];
        echo "Bienvenido, $nombre_usuario";
        echo "<button onclick='location.href='\"cerrarsesion.php\"' class='sesion'>Cerrar
        Sesión</button>";
    } else {
        echo "No has iniciado sesión.";
        echo "<button
        onclick='location.href='\"https://localhost/codigos/PantallaInicioSesion2.php\"'
        class='sesion'>Iniciar Sesión</button>";
    }
    ?>
</div>
<div class="titulo">
<?php
    echo "PLANERASMUS";
    ?>
</div>
</header>
<h2>Añada sus valoraciones</h2>

<form method="post" action="añadir_valoracionesBE.php">
<label for="Coste">Cómo valora el coste de vida?</label><br>
<select id="Coste" name="Coste">
<?php
    $i = 0;
    while($i<=10){
        echo "<option>$i</option>";
        $i++;
    }
    ?>
</select><br><br>

<label for="Transporte">Cómo valora el transporte de la ciudad?</label><br>
<select id="Transporte" name="Transporte">
<?php
    $i = 0;
    while($i<=10){
        echo "<option>$i</option>";
        $i++;
    }
    ?>
</select><br><br>

<label for="Ambiente">Cómo valora el ambiente de la ciudad?</label><br>
<select id="Ambiente" name="Ambiente">
<?php
    $i = 0;
    while($i<=10){
        echo "<option>$i</option>";
        $i++;
    }
    ?>
</select><br><br>

<label for="Comda">Cómo valora la comda de la ciudad?</label><br>
<select id="Comda" name="Comda">
<?php
    $i = 0;
    while($i<=10){
        echo "<option>$i</option>";
        $i++;
    }
    ?>
</select><br><br>

<label for="Cultura">Cómo valora la cultura de la ciudad?</label><br>
<select id="Cultura" name="Cultura">
<?php
    $i = 0;
    while($i<=10){
        echo "<option>$i</option>";
        $i++;
    }
    ?>
</select><br><br>

<label for="Clima">Cómo valora el clima de la ciudad?</label><br>
<select id="Clima" name="Clima">
<?php
    $i = 0;
    while($i<=10){
        echo "<option>$i</option>";
        $i++;
    }
    ?>
</select><br><br>

<label for="Dimensiones">Cómo valora el tamaño de la ciudad?</label><br>
<select id="Dimensiones" name="Dimensiones">
<?php
    $i = 0;
    while($i<=10){
        echo "<option>$i</option>";
        $i++;
    }
    ?>
</select><br><br>

<input type="submit" name="Valorar" value="Valorar"><br><br>
</form>
<div class="button-container">
<!-- Botón "Atras" -->
<a button onclick="volverAtras()" class="button button-back">Atras</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>

```

Añadir_valoracionesBE.php Funcionalidad Backend:

```
<?php
session_start();

if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $coste = test_input($_POST["Coste"]);
    $transporte = test_input($_POST["Transporte"]);
    $ambiente = test_input($_POST["Ambiente"]);
    $comida = test_input($_POST["Comida"]);
    $cultura = test_input($_POST["Cultura"]);
    $clima = test_input($_POST["Clima"]);
    $dimensiones = test_input($_POST["Dimensiones"]);

    if(isset($_SESSION['ciudad'])) {
        // Recuperar el error
        $ciudad = $_SESSION['ciudad'];
    }
    if(isset($_SESSION['usuario'])) {
        // Recuperar el nombre de usuario de la sesión
        $nombre_usuario = $_SESSION['usuario'];
        $seguro = $conn->prepare("Select * from valoraciones where email = ? and id_ciudad =
?");
        $seguro->bind_param("si", $nombre_usuario,$ciudad);
        $seguro->execute();
        $result = $seguro->get_result();
        $num_filas = $result->num_rows;
        if($num_filas == 0){
            $seguro = $conn->prepare("Insert into valoraciones (email,id_ciudad) Values
(?,?);");
            $seguro->bind_param("si", $nombre_usuario,$ciudad);
            $result = $seguro->execute();
        }
        $seguro = $conn->prepare("Update valoraciones
SET coste_vida = ?, transporte = ?, ambiente = ?, comida
= ?, cultura = ?, clima = ?, dimensiones = ?
where email = ? and id_ciudad = ?;");
        $seguro->bind_param("iiiiisi", $coste, $transporte, $ambiente, $comida, $cultura,
$clima, $dimensiones, $nombre_usuario,$ciudad);
        $result = $seguro->execute();
        $seguro->close();
        $conn->close();
        header("Location: https://localhost/codigos/devolverValoracionesF.php");
        exit();
    }
    else{
        header("Location: https://localhost/codigos/devolverValoracionesF.php");
        exit();
    }
}

?>
```

Acceso_chat.php Estilo con CSS:

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Acceso a los chats</title>
</head>
<style>
body {
font-family: Arial, sans-serif;
background-color: #ecf0f1;
margin: 0;
padding: 0;
}
header {
background-color: #34495e;
height: 80px;
margin-bottom: 0;
top: 0;
width: 100%;
}
h2 {
text-align: center;
font-size: 40px;
}
.boton {
display: inline-block;
padding: 10px 20px;
margin: 10px;
font-size: 18px;
text-align: center;
text-decoration: none;
border: 2px solid #4CAF50;
color: #4CAF50;
border-radius: 8px;
transition: background-color 0.3s;
background-color: white;
}

.boton:hover {
background-color: #4CAF50;
color: white;
}
.marco {
border: 2px solid #ccc;
padding: 20px;
border-radius: 10px;
width: fit-content;
margin: 0 auto;
background-color: rgba(255, 255, 255, 0.9);
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
.texto {
text-align: center;
font-weight: bold;
font-size: 25px;
}
body {
background-color: #f2f2f2;
}
.button-container {
position: fixed;
top: 10px;
right: 10px;
z-index: 9999;
}
.button {
padding: 10px 20px;
margin-left: 10px;
background-color: #5c9ded;
color: #fff;
border: none;
border-radius: 5px;
cursor: pointer;
}
.button-back {
background-color: #e74c3c;
}
.ultimo {
display: block;
}
.grupo, .privados {
margin-top: 10px;
margin-bottom: 20px;
}
.titulo {
top: 5px;
font-weight: bold;
font-size: 35px;
text-align: center;
color: #B6FDFD;
}
.usuario, .sesion {
color: #FF0616;
text-decoration: none;
margin-left: 20px;
}
.usuario {
font-size: 17px;
}
.color {
color: #B6FDFD;
}
</style>

```

Acceso_chat.php Funcionalidad con HTML y PHP:

```
<body>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>
<?php
// Iniciar la sesión
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;

?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo '<button onclick="location.href=\'cerrarsesion.php\'" class="sesion">Cerrar
Sesión</button>';
} else {
    echo "No has iniciado sesión.";
    echo '<button
onclick="location.href=\'https://localhost/codigos/PantallaInicioSesion2.php\'"
class="sesion">Iniciar Sesión</button>';
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
</header>
<h2> Bienvenido a los Chat</h2>
<div class="marco">
    <div class= "texto">
        <?php
        echo "Chats de Grupo:";
        ?>
    </div>
    <div class = "grupo">
        <a href="devolver_mensajes.php?chat=ciudad" class="boton">Accede al chat de tu ciudad
destino</a>;
        <a href="devolver_mensajes.php?chat=universidad0" class="boton">Accede al chat de tu
universidad origen</a>;
        <a href="devolver_mensajes.php?chat=universidadD" class="boton">Accede al chat de tu
universidad destino</a>;
    </div>
    <div class= "texto">
        <?php
        echo "Chats Privados:";
        ?>
    </div>
    <div class = "privados">
        <a href="chats_privados.php" class="boton ultimo">Mensajes privados con otros
usuarios</a>;
    </div>
</div>

<div class="button-container">
<!-- Botón "Atrás" -->
<a button onclick="volverAtras()" class="button button-back">Atras</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>

</body>
```

Devolver_mensajes.php Estilo con CSS:

```

<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
body {
font-family: Arial, sans-serif;
background-color: #ecf0f1;
margin: 0;
padding: 0;
}
header {
background-color: #34495e;
height: 80px;
margin-bottom: 0;
top: 0;
width: 100%;
}
h2 {
color: #333;
text-align: center;
margin-top: 10;
}
.container {
display: flex;
flex-direction: column;
height: 700px;
background-color: #5C5CCC;
}
.comments-container {
display: flex;
flex-direction: column;
flex-grow: 1;
overflow-y: auto;
border: 1px solid #ccc;
padding: 10px;
margin-bottom: 50px;
width: 80%;
align-self: center;
background-color: white;
}
.comment {
margin-bottom: 20px;
padding: 10px;
border: 1px solid #ccc;
border-radius: 5px;
background-color: #f9f9f9;
border-bottom: 1px solid #ccc;
max-width: 80%;
}
.comment-right {
align-self: flex-end;
background-color: #d9ead3;
}
.comment-left {
align-self: flex-start;
}
.comment .info {
font-size: 14px;
color: #8080;
margin-bottom: 5px;
}
.comment .text {
font-size: 16px;
line-height: 1.2;
}
form {
border-radius: 10px;
padding: 10px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
margin-bottom: 20px;
}
label {
display: block;
font-weight: bold;
margin-bottom: 5px;
}
input[type="text"] {
width: calc(100% - 20px);
padding: 10px;
border: 1px solid #ccc;
border-radius: 5px;
font-size: 16px;
margin-bottom: 10px;
}
input[type="text"]:focus {
outline: none;
border-color: #5c9ded;
box-shadow: 0 0 4px rgba(92, 157, 237, 0.5);
}
input[type="submit"] {
background-color: #5c9ded;
color: #fff;
border: none;
border-radius: 5px;
padding: 10px 20px;
font-size: 16px;
cursor: pointer;
}
input[type="submit"]:hover {
background-color: #4ab8c4;
}
.button-container {
position: fixed;
top: 10px;
right: 10px;
z-index: 9999;
}
.button {
padding: 10px 20px;
margin-left: 10px;
background-color: #5c9ded;
color: #fff;
border: none;
border-radius: 5px;
cursor: pointer;
}
.button-back {
background-color: #e74c3c;
}
.barra {
position: fixed;
bottom: 0;
background-color: #34495e;
width: 100%;
}
.titulo {
top: 5px;
font-weight: bold;
font-size: 16px;
text-align: center;
color: #86f0ff;
}
.usuario, .sesion {
color: #f9b84d;
text-decoration: none;
margin-left: 20px;
}
.usuario {
font-size: 17px;
}
.color {
color: #86f0ff;
}
</style>

```

Devolver_mensajes.php Funcionalidad con HTML y PHP:

```

<body>
<script>
function volverAtras() {
    window.history.back();
}
</script>
<header>
</header>
<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
// Iniciar la sesión
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;
?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo "<button onclick='location.href='\"cerrarsesion.php\"' class='session'>Cerrar
Sesión</button>";
} else {
    echo "No has iniciado sesión.";
    echo "<button
onclick='location.href='\"https://localhost/codigos/PantallaInicioSesion2.php\"'
class='session'>Iniciar Sesión</button>";
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
<?php
$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";
$ciudad = "";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");
?>
<div class="container">
<?php
$id_chat = 0;
if (isset($_GET['chat'])) {
    $chat = test_input($_GET['chat']);
    $usuario = $_SESSION['usuario'];
    $id_chat = 0;
    $cabecera = "";
    $nombre = "";
    if($chat=="ciudad" || $chat=="universidad" || $chat=="universidadD") {
        if($chat=="ciudad") {
            $seguro = $conn->prepare("SELECT c.id_chat as id
FROM chat c inner join pertenece p inner join
referente r on
c.id_chat = p.id_chat and r.id_chat = c.id_chat
where p.email_usuario = ? and r.tipo_referencia =
'ciudad'");
            $seguro2 = $conn->prepare("SELECT nombre
From ciudad
where id_ciudad = (SELECT ciudad_destino FROM
usuario where email = ?)");
            $cabecera = "de la ciudad destino: ";
        }
        else if($chat == 'universidadD') {
            $seguro = $conn->prepare("SELECT c.id_chat as id
FROM chat c inner join pertenece p inner join
referente r on
c.id_chat = p.id_chat and r.id_chat = c.id_chat
where p.email_usuario = ? and r.tipo_referencia =
'universidad' and r.tipo_universidad = 'origen'");
            $seguro2 = $conn->prepare("SELECT nombre
From universidad
where id_universidad = (SELECT universidad_origen
FROM usuario where email = ?)");
            $cabecera = "de la universidad origen: ";
        }
        else if($chat == 'universidad') {
            $seguro = $conn->prepare("SELECT c.id_chat as id
FROM chat c inner join pertenece p inner join
referente r on
c.id_chat = p.id_chat and r.id_chat = c.id_chat
where p.email_usuario = ? and r.tipo_referencia =
'universidad' and r.tipo_universidad = 'destino'");
            $seguro2 = $conn->prepare("SELECT nombre
From universidad
where id_universidad = (SELECT universidad_destino
FROM usuario where email = ?)");
            $cabecera = "de la universidad destino: ";
        }
        $seguro->bind_param("s", $usuario);
        $seguro->execute();
        $result = $seguro->get_result();
        $fila = $result->fetch_assoc();
        $nombre = $fila['nombre'];
        $seguro->bind_param("s", $usuario);
        $seguro->execute();
        $result = $seguro->get_result();
        $fila = $result->fetch_assoc();
        $id_chat = $fila['id'];
    }
    else {
        $id_chat = $chat;
        $cabecera = "del chat privado";
    }
}
?>
<div class="container">
<?php
echo "<h2>Mensajes del chat <span>Cabecera <span>nombre/h2";
?>
<div class="comments-container">
<?php
$seguro = $conn->prepare("SELECT * FROM mensaje where id_chat = ?");
$seguro->bind_param("i", $id_chat);
$seguro->execute();
$result = $seguro->get_result();
while(($fila = $result->fetch_assoc())) {
    $nombre_usuario = $fila['email_usuario'];
    $fecha = $fila['fecha'];
    $texto = $fila['texto'];
    $comentario_propio = ($nombre_usuario == $_SESSION['usuario']);
    $clase_comentario = ($comentario_propio ? 'comment-right' : 'comment-left');
}
?>
<div class="comment">?php echo $clase_comentario; ?>
<div class="info">?php echo date("d/m/Y H:i", strtotime($fecha)); ?> - Usuario
<?php echo $nombre_usuario; ?></div>
<div class="text">?php echo $texto; ?></div>
</div>
<?php
?>
</div>
</div>
<div class="barra">
<form method="post" action="introducir_mensaje.php">
<label for="mensaje">Escriba:/label>
<input type="text" id="mensaje" name="mensaje" required=>br/>
<input type="hidden" name="chat" value="<?php echo $id_chat; ?>">
<input type="submit" name="EnviarMensaje" value="Enviar">>br/>
</form>
</div>
<div class="button-container">
<div class="boton Atras">
<a href="#">
<a href="#">
</div>
</div>
</body>
</html>

```

Introducir_mensaje.php Funcionalidad Backend con PHP:

```
<?php
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $mensaje = test_input($_POST['mensaje']);
    $id_chat = test_input($_POST['chat']);
    if(isset($_SESSION['usuario'])) {
        // Recuperar el nombre de usuario de la sesión
        $nombre_usuario = $_SESSION['usuario'];
        $seguro = $conn->prepare("INSERT INTO mensaje (texto,email_usuario,id_chat) VALUES
(?,?,?);");
        $seguro->bind_param("ssi", $mensaje,$nombre_usuario,$id_chat);
        $result = $seguro->execute();
        $seguro->close();
        $conn->close();
        if($result == TRUE){
            echo "Comentario realizado con éxito";
            header("Location: $url_anterior");
            exit();
        }
        else{
            echo "Ha surgido algún problema al añadir comentario";
            header("Location: $url_anterior");
            exit();
        }
    }
    else{
        echo "Por favor inicie sesión y vuelva a interarlo";
        header("refresh:3; url=https://localhost/codigos/PantallaInicioSesion2.php");
        exit();
    }
}

?>
```

Chats_privados.php Estilo con CSS:

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
body {
font-family: Arial, sans-serif;
background-color: #ecf0f1;
margin: 0;
padding: 0;
}
header{
background-color: #34495e;
height: 80px;
margin-bottom: 0;
top: 0;
width: 100%;
}
form {
text-align: left;
margin-top: 20px;
}
form.centrado{
text-align: center;
}

input[type="text"] {
padding: 10px;
border: 1px solid #ccc;
border-radius: 5px;
width: 300px;
box-sizing: border-box;
}
input[type="submit"] {
padding: 10px 20px;
background-color: #5c9ded;
color: #fff;
border: none;
border-radius: 5px;
cursor: pointer;
}
input[type="submit"].chat {
width: 500px;
height: 75px;
}
.button-container {
position: fixed;
top: 10px;
right: 10px;
z-index: 9999;
}
.button {
padding: 10px 20px;
margin-left: 10px;
background-color: #5c9ded;
color: #fff;
border: none;
border-radius: 5px;
cursor: pointer;
}
.button-back {
background-color: #e74c3c;
}
.marco {
border: 2px solid #ccc;
padding: 20px;
border-radius: 10px;
width: 1000px;
margin: 50px auto;
position: center;
background-color: rgba(255, 255, 255, 0.9);
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
text-align: left;
}
.texto{
font-weight: bold;
font-size: 20px;
}
.texto2{
font-weight: bold;
font-size: 20px;
text-align: center;
}
.titulo{
top: 5px;
font-weight: bold;
font-size: 35px;
text-align: center;
color: #B6F0FF;
}
.usuario, .sesion {
color: #FF0616;
text-decoration: none;
margin-left: 20px;
}
.usuario {
font-size: 17px;
}
.color{
color: #B6F0FF;
}
</style>

```

Chats_privados.php Funcionalidad con HTML y PHP:

```
<body>
<script>
    function volverAtras() {
        window.history.back();
    }
</script>
<header>
<?php
// Iniciar la sesión
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://".$_SERVER['HTTP_HOST'].$_SERVER['REQUEST_URI'];
$_SESSION['anterior'] = $current_url;

?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo "<button onclick='location.href=\"'.\"cerrarsesion.php'\" class='sesion'>Cerrar Sesión</button>";
} else {
    echo "No has iniciado sesión.";
    echo "<button onclick='location.href=\"'.\"https://localhost/codigos/PantallaInicioSesion2.php'\" class='sesion'>Iniciar Sesión</button>";
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>
<?php
$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");
?>
</header>
<div class="marco">
</form>
<form method="post" action="explorar_chats.php" class="centrado">
    <input type="submit" name="Buscar" value="Acceda a sus chats" class="chat"><br><br>
</form>
</div>
<div class="texto2">
<?php
echo "0";
?>
</div>
<div class="marco">
<div class="texto">
<?php
echo "Busque usuarios por alguno de los siguientes filtros:";
?>
</div>

<form method="post" action="buscar_usuario.php">
    <label for="mensaje">Busque por nombre:</label>
    <input type="text" id="nombre" name="nombre" required>
    <input type="submit" name="Buscar" value="Buscar"><br><br>
</form>
<form method="post" action="buscar_usuario.php">
    <label for="mensaje">Busque usuarios de una ciudad:</label>
    <select id="ciudad" name="ciudad" required>
        <?php
        $mysqli = mysqli_connect("localhost:3306", "Plan Erasmus", "ErasmusBudapest", "planerasmus");
        $mysqli->set_charset("utf8");
        $resultado = mysqli_query($mysqli, "select * from ciudad order by nombre");
        while($fila = $resultado->fetch_assoc()){
            $id = $fila['id_ciudad'];
            $nombre = $fila['nombre'];
            echo "<option value='$id'>$nombre</option>";
        }
        ?>
    </select>
    <input type="submit" name="Buscar" value="Buscar"><br><br>
</form>
<form method="post" action="buscar_usuario.php">
    <label for="mensaje">Busque usuarios de una universidad origen:</label>
    <select id="universidad0" name="universidad0" required>
        <?php
        $mysqli = mysqli_connect("localhost:3306", "Plan Erasmus", "ErasmusBudapest", "planerasmus");
        $mysqli->set_charset("utf8");
        $resultado = mysqli_query($mysqli, "select * from universidad order by nombre");
        while($fila = $resultado->fetch_assoc()){
            $id = $fila['id_universidad'];
            $nombre = $fila['nombre'];
            echo "<option value='$id'>$nombre</option>";
        }
        ?>
    </select>
    <input type="submit" name="Buscar" value="Buscar"><br><br>
</form>
<form method="post" action="buscar_usuario.php">
    <label for="mensaje">Busque usuarios de una universidad destino:</label>
    <select id="universidad0" name="universidad0" required>
        <?php
        $mysqli = mysqli_connect("localhost:3306", "Plan Erasmus", "ErasmusBudapest", "planerasmus");
        $mysqli->set_charset("utf8");
        $resultado = mysqli_query($mysqli, "select * from universidad order by nombre");
        while($fila = $resultado->fetch_assoc()){
            $id = $fila['id_universidad'];
            $nombre = $fila['nombre'];
            echo "<option value='$id'>$nombre</option>";
        }
        ?>
    </select>
    <input type="submit" name="Buscar" value="Buscar"><br><br>
</form>
</div>
<div class="button-container">
<!-- Botón "Atrás" -->
<a button onclick="volverAtras()" class="button button-back">Atrás</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
</body>
</html>
```

Explorar_chats.php Estilo con CSS:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #ecf0f1;
    margin: 0;
    padding: 0;
  }
  header{
    background-color: #34495e;
    height: 80px;
    margin-bottom: 0;
    top: 0;
    width: 100%;
  }
  .usuarios-container {
    height: 800px;
    overflow-y: auto;
    border: 3px solid #ccc;
    padding: 10px;
    margin-bottom: 20px;
    text-align: center;
    justify-content: center;
    max-width: 100%;
    margin: auto;
    background-color: #ecf0f1;
  }
  .usuario-marco {
    border: 3px solid #ddd;
    padding: 10px;
    margin-bottom: 20px;
    text-align: center;
    color: #BA7007;
    align-items: center;
    background-color: #CDFCFB;
    max-width: 500px;
    margin: auto;
    border-color: #485A5C;
  }
  .button-container {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
  }
  .button {
    padding: 10px 20px;
    margin-left: 10px;
    background-color: #5c9ded;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  .button-back {
    background-color: #e74c3c;
  }
  .titulo{
    top: 5px;
    font-weight: bold;
    font-size: 35px;
    text-align: center;
    color: #B6FDFD;
  }
  .usuario, .sesion {
    color: #FF0616;
    text-decoration: none;
    margin-left: 20px;
  }
  .usuario {
    font-size: 17px;
  }
  .color{
    color: #B6FDFD;
  }
</style>
```

Explorar_chats.php Funcionalidad con HTML y PHP:

```

</body>
</header>
<script>
function volverAtras() {
    window.history.back();
}
</script>
<div class="button-container">
<!-- Botón "Atrás" -->
<a button onclick="volverAtras()" class="button button-back">Atras</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
<?php
// Iniciar la sesión
session_start();
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;

?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo '<button onclick="location.href=\'cerrarsesion.php\'" class="sesion">Cerrar
Sesión</button>';
} else {
    echo "No has iniciado sesión.";
    echo '<button
onclick="location.href=\'https://localhost/codigos/PantallaInicioSesion2.php\'"
class="sesion">Iniciar Sesión</button>';
}
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMUS";
?>
</div>

<?php
$servername = "localhost:3306";
$username = "Plan_Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");

echo "</header>";
if(isset($_SESSION['usuario'])) {
    // Recuperar el nombre de usuario de la sesión
    $nombre_usuario = $_SESSION['usuario'];
    $seguro = $conn->prepare("SELECT c.id_chat
FROM chat c inner join pertenece p
on c.id_chat = p.id_chat
where c.privado = true and p.email_usuario = ?;");
    $seguro->bind_param("s", $nombre_usuario);
    $seguro->execute();
    $result = $seguro->get_result();
    echo '<div class="usuarios-container">';
    if($result->num_rows!>0){
        while($fila = $result->fetch_assoc()){
            $id_chat = $fila['id_chat'];
            $seguro2 = $conn->prepare("SELECT *
FROM pertenece p inner join usuario u
on p.email_usuario = u.email
where id_chat = ? and p.email_usuario != ?;");
            $seguro2->bind_param("is", $id_chat,$nombre_usuario);
            $seguro2->execute();
            $result2 = $seguro2->get_result();
            $fila2 = $result2->fetch_assoc();
            $email = $fila2['email'];
            $nombre = $fila2['nombre'];
            echo '<div class="usuario-marco">';
            echo "NOMBRE USUARIO: $email";
            echo "<br>";
            echo "NOMBRE: $nombre";
            echo "<br>";
        }
        echo '</div>';
    }
}
$seguro2->close();
$seguro->close();
$conn->close();
echo '</div>';
}
else{
    echo "INICIE SESIÓN PARA VER SUS CHATS";
}
?>

</body>
</html>

```

Buscar_usuario.php Estilo con CSS:

```
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
body {
  font-family: Arial, sans-serif;
  background-color: #ecf0f1;
  margin: 0;
  padding: 0;
}
header{
  background-color: #34495e;
  height: 80px;
  margin-bottom: 0;
  top: 0;
  width: 100%;
}
.usuarios-container {
  height: 800px;
  overflow-y: auto;
  border: 3px solid #ccc;
  padding: 10px;
  margin-bottom: 20px;
  text-align: center;
  justify-content: center;
  max-width: 100%;
  margin: auto;
  background-color: #ecf0f1;
}
.usuario-marco {
  border: 3px solid #ddd;
  padding: 10px;
  margin-bottom: 20px;
  text-align: center;
  color: #BA7007;
  align-items: center;
  background-color: #CDFCFB;
  max-width: 500px;
  margin: auto;
  border-color: #485A5C;
}
.button-container {
  position: fixed;
  top: 10px;
  right: 10px;
  z-index: 9999;
}
.button {
  padding: 10px 20px;
  margin-left: 10px;
  background-color: #5c9ded;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
.button-back {
  background-color: #e74c3c;
}
.titulo{
  top: 5px;
  font-weight: bold;
  font-size: 35px;
  text-align: center;
  color: #B6FDFD;
}
.usuario, .sesion {
  color: #FF0616;
  text-decoration: none;
  margin-left: 20px;
}
.usuario {
  font-size: 17px;
}
.color{
  color: #B6FDFD;
}
</style>
```

Buscar_usuario.php Funcionalidad con HTML y PHP:

```

<script>
function volverAtras() {
    window.history.back();
}
</script>
<body>
<header>
<div class="button-container">
<!-- Botón "Atrás" -->
<a button onClick="volverAtras()" class="button button-back">Atras</button></a>
<!-- Botón "Home" -->
<a href="PantallaInicio.php" class="button">Home</a>
</div>
</php>
<!-- Iniciar la sesión -->
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
session_start();
$nombre_usuario = "";
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
$current_url = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
$_SESSION['anterior'] = $current_url;
?>
<div class="color">
<?php
// Verificar si existe la variable de sesión 'usuario'
if(isset($_SESSION['usuario'])) {
    $nombre_usuario = $_SESSION['usuario'];
    echo "Bienvenido, $nombre_usuario";
    echo <button onClick="location.href='cerrarsesion.php'" class="sesion">Cerrar
Sesión</button>;
} else {
    echo "No has iniciado sesión.";
    echo <button
onClick="location.href='https://localhost/codigos/PantallaInicioSesion2.php'"
class="sesion">Iniciar Sesión</button>;
?>
</div>
<div class="titulo">
<?php
echo "PLANERASMIUS";
?>
</div>
<?php
$servername = "localhost:3306";
$username = "Plan_Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmius";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");
echo </header>;

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nombre_aux = $_POST["nombre"] ?? null;
    $id_ciudad = $_POST["ciudad"] ?? null;
    $id_universidad = $_POST["universidad"] ?? null;
    $id_universidad0 = $_POST["universidad0"] ?? null;
    $result = null;
    if($nombre_aux != null){
        $nombre = "%".$nombre_aux."%";
        $seguro = $conn->prepare("SELECT u.nombre as nombre, email, c.nombre as ciudad
from usuario u inner join ciudad c
on u.ciudad_destino = c.id_ciudad
where (u.nombre like ? or email like ?) and u.email != ?;");
        $seguro->bind_param("sss", $nombre, $nombre, $nombre_usuario);
        $seguro->execute();
        $result = $seguro->get_result();
    }
    else if($id_ciudad != null){
        $seguro = $conn->prepare("SELECT u.nombre as nombre, email, c.nombre as ciudad
from usuario u inner join ciudad c
on u.ciudad_destino = c.id_ciudad
where c.id_ciudad = ? and u.email != ?;");
        $seguro->bind_param("is", $id_ciudad, $nombre_usuario);
        $seguro->execute();
        $result = $seguro->get_result();
    }
    else if($id_universidad != null){
        $seguro = $conn->prepare("SELECT u.nombre as nombre, email, c.nombre as ciudad
from usuario u inner join ciudad c inner join universidad s
on u.ciudad_destino = c.id_ciudad and u.universidad_origen = s.id_universidad
where s.id_universidad = ? and u.email != ?;");
        $seguro->bind_param("is", $id_universidad, $nombre_usuario);
        $seguro->execute();
        $result = $seguro->get_result();
    }
    else if($id_universidad0 != null){
        $seguro = $conn->prepare("SELECT u.nombre as nombre, email, c.nombre as ciudad
from usuario u inner join ciudad c inner join universidad s
on u.ciudad_destino = c.id_ciudad and u.universidad_destino = s.id_universidad
where s.id_universidad = ? and u.email != ?;");
        $seguro->bind_param("is", $id_universidad0, $nombre_usuario);
        $seguro->execute();
        $result = $seguro->get_result();
    }
}
if ($result->num_rows == 0){
    echo "No hay usuarios que correspondan con su criterio de búsqueda";
}
echo <div class="usuarios-container">;
while($fila = $result->fetch_assoc()){
    $nombreU = $fila['nombre'];
    $email = $fila['email'];
    $ciudad = $fila['ciudad'];
    echo <div class="usuario-marco">;
    echo "NOMBRE USUARIO: $email";
    echo <br>;
    echo "NOMBRE: $nombreU";
    echo <br>;
    echo "CIUDAD DESTINO: $ciudad";
    echo <br>;
?>
<div>
<a href="crear_chat.php?email=?php echo $email"><?php echo "Escribir";?>
</a>;
</div>
<?php
echo </div>;
}
echo </div>;
} else {
    echo "error";
}
?>
</body>
</html>

```

Crear_chats.php Funcionalidad Backend:

```
<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

session_start();
$nombre_usuario = "";
$url_anterior = "";
if(isset($_SESSION['anterior'])) {
    $url_anterior = $_SESSION['anterior'];
}
if(isset($_SESSION['usuario'])) {
    // Recuperar el nombre de usuario de la sesión
    $nombre_usuario = $_SESSION['usuario'];
}
$servername = "localhost:3306";
$username = "Plan Erasmus";
$password = "ErasmusBudapest";
$dbname = "planerasmus";
$ciudad = "";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8");

if (isset($_GET['email']) && isset($_SESSION['usuario'])) {
    $email = $_GET['email'];
    $seguro = $conn->prepare("SELECT c.id_chat
        FROM chat c inner join pertenece p
        on c.id_chat = p.id_chat
        where p.email_usuario = ? and c.privado = true;");
    $seguro->bind_param("s", $nombre_usuario);
    $seguro->execute();
    $result = $seguro->get_result();
    $chats_usuario = array();
    while($fila = $result->fetch_assoc()){
        $chats_usuario[] = $fila['id_chat'];
    }
    $seguro = $conn->prepare("SELECT c.id_chat
        FROM chat c inner join pertenece p
        on c.id_chat = p.id_chat
        where p.email_usuario = ? and c.privado = true;");
    $seguro->bind_param("s", $email);
    $seguro->execute();
    $result = $seguro->get_result();
    $encontrado = false;
    $longitud = count($chats_usuario);
    while($fila = $result->fetch_assoc() and $encontrado == false){
        $id_chat = $fila['id_chat'];
        for ($i = 0; $i < $longitud and $encontrado == false; $i++){
            if($chats_usuario[$i] == $id_chat){
                $encontrado = true;
            }
        }
    }
    if($encontrado == true){
        header("Location:devolver_mensajes.php?chat=$id_chat");
    }
    else{
        $sql = "INSERT INTO chat (num_participantes, privado) VALUES (2, true)";
        if ($conn->query($sql) === TRUE) {
            $id_chat = $conn->insert_id;
            $seguro = $conn->prepare("Insert into pertenece (email_usuario,id_chat) Values
            (?,?);");
            $seguro->bind_param("si", $nombre_usuario,$id_chat);
            $result = $seguro->execute();
            $seguro = $conn->prepare("Insert into pertenece (email_usuario,id_chat) Values
            (?,?);");
            $seguro->bind_param("si", $email,$id_chat);
            $result = $seguro->execute();
            header("Location:devolver_mensajes.php?chat=$id_chat");
        }
        else{
            echo "No se ha podido crear el chat";
            header("Location:$url_anterior");
        }
    }
}
?>
```

Inserción inicial de datos en la Base de Datos:

```
public boolean openConnection() {  
    //Metodo para establecer conexion con la base de datos, primero se cargan los  
drivers  
    //para luego conectarse a la base de datos con un usuario y password  
    String driv = "com.mysql.cj.jdbc.Driver";  
    try {  
        Class.forName(driv);  
        String serverAdress = "localhost:3306";  
        String db = "planerasmus";  
        String user = "Plan Erasmus";  
        String pass = "ErasmusBudapest";  
        String url = "jdbc:mysql://" + serverAdress + "/" + db;  
        conn = DriverManager.getConnection(url, user, pass);  
        return true;  
    }  
    catch (ClassNotFoundException e) {return false; }  
    catch (SQLException e) {return false; }  
}  
  
public boolean closeConnection() {  
    //Metodo que cierra la conexion con la base de datos  
    try { conn.close(); }  
    catch (SQLException e) {return false;}  
    return true;  
}
```

```

public void ejecutar() {

    String inputFile = "C:\\Users\\Luis Miranda\\Desktop\\Informatica +
ade\\TFG\\Universidades europa.txt"; // Nombre del archivo de entrada
    String outputFile = "C:\\Users\\Luis Miranda\\Desktop\\Informatica +
ade\\TFG\\Ciudades europa.txt"; // Nombre del archivo de salida

    BufferedReader bf = null;
    BufferedWriter wr = null;
    try {

        bf = new BufferedReader(new InputStreamReader(new FileInputStream(inputFile),
"UTF-8"));
        wr = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(outputFile),
"UTF-8"));

        String tupla = bf.readLine();
        while(tupla!=null) {
            String cambio1[] = tupla.split("-");
            String cambio2[] = cambio1[cambio1.length-1].split(",");
            wr.append(cambio2[0]);
            wr.append(" ");
            wr.append(cambio2[1]);
            wr.append("\n");
            tupla = bf.readLine();
        }
        if(bf != null)
            bf.close();
        if(wr != null)
            wr.close();

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        try {
            if(bf != null)
                bf.close();
            if(wr != null)
                wr.close();
        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
}
}

```

```

public void comprimir() {
    String inputFile = "C:\\Users\\Luis Miranda\\Desktop\\Informatica +
ade\\TFG\\Ciudades europa.txt"; // Nombre del archivo de entrada
    String outputFile = "C:\\Users\\Luis Miranda\\Desktop\\Informatica +
ade\\TFG\\Ciudades europa Comprimido.txt"; // Nombre del archivo de salida

    BufferedReader bf = null;
    BufferedWriter wr = null;

    try {
        bf = new BufferedReader(new InputStreamReader(new FileInputStream(inputFile),
"UTF-8"));
        wr = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(outputFile),
"UTF-8"));

        String tupla = bf.readLine();
        String ciudades [] = new String [600];

        while(tupla != null) {
            String separ[] = tupla.split(" ");
            int i = 0;
            boolean encontrado = false;
            while(ciudades[i]!=null && !encontrado) {
                if (ciudades[i].equals(separ[1]))
                    encontrado = true;
                i++;
            }
            if(!encontrado) {
                wr.append(separ[1]);
                wr.append(",");
                int j = 3;
                while(j<separ.length) {
                    wr.append(separ[j]);
                    wr.append(" ");
                    j++;
                }
                wr.append("\n");
                ciudades[i] = String.copyValueOf(separ[1].toCharArray());
            }
            tupla = bf.readLine();
        }
        if(bf != null)
            bf.close();
        if(wr != null)
            wr.close();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        try {
            if(bf != null)
                bf.close();
            if(wr != null)
                wr.close();
        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
}
}

```

```

public void insertarCiudades() {
    String inputFile = "C:\\Users\\Luis Miranda\\Desktop\\Informatica +
ade\\TFG\\Ciudades europa Comprimido.txt"; // Nombre del archivo de entrada
    BufferedReader bf = null;
    String query = "INSERT INTO ciudad (nombre, pais) VALUES (?,?)";
    PreparedStatement pst = null;

    try {
        bf = new BufferedReader(new InputStreamReader(new FileInputStream(inputFile),
"UTF-8"));
        String tupla = bf.readLine();
        conn.setAutoCommit(false);
        pst = conn.prepareStatement(query);

        while(tupla != null) {
            String aux[] = tupla.split(",");
            pst.setString(1, aux[0]);
            pst.setString(2, aux[1]);
            pst.executeUpdate();
            tupla = bf.readLine();
        }
        if(bf != null)
            bf.close();
        conn.commit();
        pst.close();

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        try {
            if(bf != null)
                bf.close();
            conn.rollback();
            if(pst!=null)
                pst.close();

        } catch (Exception e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
}

```

```

public void insertarUniversidades() {
    String inputFile = "C:\\Users\\Luis Miranda\\Desktop\\Informatica +
ade\\TFG\\Universidades europa.txt";
    BufferedReader bf = null;
    PreparedStatement pst = null;
    PreparedStatement pst2 = null;
    ResultSet rst = null;
    try {
        bf = new BufferedReader(new InputStreamReader(new FileInputStream(inputFile),
"UTF-8"));
        String query = "SELECT id_ciudad FROM ciudad where nombre = ?";
        String query2 = "INSERT INTO universidad (nombre,id_ciudad) VALUES (?,?)";
        conn.setAutoCommit(false);
        pst = conn.prepareStatement(query);
        pst2 = conn.prepareStatement(query2);

        String tupla = bf.readLine();
        while(tupla != null) {
            String aux[] = tupla.split(" - ");
            String ciudad[] = aux[1].split(",");
            pst.setString(1, ciudad[0]);
            rst = pst.executeQuery();
            if(rst.next()) {
                int codigo_ciudad = rst.getInt("id_ciudad");
                String universidad[] = aux[0].split(" ");
                String uni = "";
                int i = 1;
                while(i<universidad.length) {
                    uni = uni + universidad[i] + " ";
                    i++;
                }
                pst2.setString(1, uni);
                pst2.setInt(2, codigo_ciudad);
                pst2.executeUpdate();
            }
            tupla = bf.readLine();
        }
        if(bf != null)
            bf.close();
        conn.commit();
        pst.close();
        pst2.close();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        try {
            if(bf != null)
                bf.close();
            conn.rollback();
            if(pst!=null)
                pst.close();
            if(pst2!=null)
                pst2.close();
        } catch (Exception e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
}
}

```

```
public void crearChatsCiudades() {
    PreparedStatement pst = null;
    PreparedStatement pst1 = null;
    PreparedStatement pst2 = null;
    int id_ciudad;
    String query = "SELECT id_ciudad FROM ciudad";
    String query2 = "Insert into chat (id_chat,num_participantes) VALUES (?,0)";
    String query3 = "Insert into referente
(id_chat,id_referencia_ciudad,tipo_referencia) VALUES (?,?,'ciudad')";
    try {
        conn.setAutoCommit(false);
        pst = conn.prepareStatement(query);
        ResultSet rst = pst.executeQuery();
        int i = 1;
        while(rst.next()) {
            pst1 = conn.prepareStatement(query2);
            pst1.setInt(1, i);
            pst1.executeUpdate();
            i++;
        }
        i = 1;
        rst = pst.executeQuery();
        while(rst.next()) {
            id_ciudad = rst.getInt("id_ciudad");
            pst2 = conn.prepareStatement(query3);
            pst2.setInt(1, i);
            pst2.setInt(2, id_ciudad);
            pst2.executeUpdate();
            i++;
        }
        conn.commit();
        pst.close();
        pst1.close();
        pst2.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```


```

public void crearChatsUniversidades() {
    PreparedStatement pst = null;
    PreparedStatement pst1 = null;
    PreparedStatement pst2 = null;
    PreparedStatement pst3 = null;
    String query = "SELECT id_chat FROM chat";
    String query1 = "SELECT id_universidad FROM universidad";
    String query2 = "Insert into chat (id_chat,num_participantes) VALUES (?,0)";
    String query3 = "Insert into referente
(id_chat,id_referencia_universidad,tipo_referencia,tipo_universidad) VALUES
(?,?, 'universidad',?)";
    int id_universidad;
    int contador_fila;
    int id1;

    try {
        conn.setAutoCommit(false);
        pst = conn.prepareStatement(query);
        ResultSet rst = pst.executeQuery();
        contador_fila = 0;
        while(rst.next()) {
            contador_fila++;
        }
        contador_fila++;
        id1=contador_fila;
        pst2 = conn.prepareStatement(query1);
        rst = pst2.executeQuery();
        while(rst.next()) {
            pst1 = conn.prepareStatement(query2);
            pst1.setInt(1, contador_fila);
            pst1.executeUpdate();
            contador_fila++;
            pst1 = conn.prepareStatement(query2);
            pst1.setInt(1, contador_fila);
            pst1.executeUpdate();
            contador_fila++;
        }
        pst2 = conn.prepareStatement(query1);
        rst = pst2.executeQuery();
        contador_fila = id1;
        while(rst.next()) {
            id_universidad = rst.getInt("id_universidad");
            pst3 = conn.prepareStatement(query3);
            pst3.setInt(1, contador_fila);
            pst3.setInt(2, id_universidad);
            pst3.setString(3, "origen");
            pst3.executeUpdate();
            contador_fila++;
            pst3 = conn.prepareStatement(query3);
            pst3.setInt(1, contador_fila);
            pst3.setInt(2, id_universidad);
            pst3.setString(3, "destino");
            pst3.executeUpdate();
            contador_fila++;
        }
        conn.commit();
        pst.close();
        pst1.close();
        pst2.close();
        pst3.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

Este documento esta firmado por



| | |
|-------------------------------|---|
| Firmante | CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES |
| Fecha/Hora | Mon Jun 03 21:39:50 CEST 2024 |
| Emisor del Certificado | EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES |
| Numero de Serie | 561 |
| Metodo | urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature) |