



CICLO DESARROLLO DE APLICACIONES WEB

Trabajo Final de Grado: KIRICASA

RICARDO SORIN ALMAJAN

Colegio Montessori

Índice

Contenido

Índice	1
Introducción	5
Objetivos específicos	6
Tecnologías Utilizadas	7
Backend.....	7
Frontend.....	8
Desarrollo de la aplicación	8
Planificación del proyecto	14
Fases principales	14
Diagrama de Gantt.....	15
Diagrama Entidad-Relación.....	16
1. Usuarios.....	16
2. Publicaciones.....	17
3. Barrios	17
4. Favoritos.....	18
5. Anuncios Vistos	18
Estructura del sitio web.....	18
Cabeecera (Header)	18
No logged	19
Logged	19
Admin	19
Homepage.....	20
Footer.....	21
Perfil.....	22
Publicaciones	24
Subida de un Anuncio	28
Búsqueda	29
Paginas Varias	30
Blog.....	30
Política de Privacidad	31

Términos y Condiciones de Uso	31
Administración	32
Plan de Empresa	33
1.-Oportunidades de negocio y necesidades demandadas.....	33
Justificación de la oportunidad de negocio	33
Descripción de la localización de la empresa.....	33
Estudio de nuestro cliente y segmentación de mercado.....	33
2. Relación de productos o servicios a prestar. Comenta los aspectos destacados acerca de estos apartados.	34
2.1 Producto/servicio	34
2.2 Precio.....	34
2.3 Promoción	35
2.4 Distribución	35
3. Constitución de la empresa	36
3.1. Tipo de sociedad elegido explicando las razones	36
3.2. Nombra la relación de trámites mercantiles, fiscales, laborales y locales que debe realizar la empresa	36
4. Recursos	37
4.1.-Recursos Materiales/Técnicos	37
4.2.-Recursos Humanos.....	38
4.3.-Recursos Económicos.....	38
Conclusiones	39
Guía de Estilos	40
Introducción.....	40
Logotipo	40
Fuentes.....	40
sans serif.....	40
Nunito ExtraBold	40
Baloo.....	41
Cabin Rounded	41
Cabecera	41
Diseño del footer	43
Login.....	44

Contraseña olvidada	46
Visualización de un anuncio.....	47
Ver perfil	48
Modificar perfil	49
Activación verificación 2 pasos.....	50
Blog	51
Términos y Condiciones.....	52
Pestaña Enviar Mensaje.....	52
Pruebas	53
Inicio de Sesión:	53
Registro	55
Búsqueda	60
Publicar Anuncios.....	61
Modificar datos usuarios	63
Modificar Publicación	64
Recuperar Contraseña	65
Manual de Usuario	66
¿Qué es Kiricasa?	66
Registro y acceso.....	66
Caso: He olvidado Mi contraseña/Modificar Contraseña	69
Cambiar Contraseña.....	69
He olvidado mi contraseña	69
Caso: Publicar Anuncio	70
Caso: Modificar Anuncio.....	73
Manual De Administrador	74
Manual De Despliegue.....	77
Despliegue en local	77
1. Requisitos del sistema.....	77
2. Configuración de base de datos	78
3. Clonar o copiar el proyecto	78
4. Generar el JAR ejecutable	78
5. Ejecutar la aplicación.....	78
6. Acceso a la plataforma	79

Despegue Online:.....	80
Opción Heroku.....	80
Opción Apache TomCat en un servidor propio o VPS.....	80
Conclusiones.....	83
Bibliografía.....	84

Introducción

La búsqueda de alojamiento, ya sea de habitaciones o pisos completos, representa una necesidad constante para muchas personas, especialmente en entornos urbanos con alta movilidad como la ciudad de Zaragoza. Estudiantes universitarios, jóvenes profesionales o personas recién llegadas a la ciudad se enfrentan frecuentemente a procesos complejos y poco optimizados a la hora de encontrar un lugar donde vivir. Plataformas consolidadas como Wallapop o Milanuncios ofrecen soluciones genéricas, pero a menudo presentan interfaces sobrecargadas, procesos poco intuitivos o un enfoque centrado en la compraventa general, más que en el alquiler de viviendas compartidas.

Kiricasa surge como una respuesta directa a esta problemática, proponiendo una plataforma web moderna, limpia y específica para el alquiler de habitaciones y pisos. A diferencia de otras soluciones, Kiricasa está pensada para ser intuitiva desde el primer uso, adaptada al público joven, accesible desde cualquier dispositivo, y centrada inicialmente en una sola ciudad para poder ofrecer una experiencia local personalizada.

Este proyecto ha sido desarrollado como Trabajo Fin de Grado en el ciclo formativo de Desarrollo de Aplicaciones Web, combinando una solución técnica basada en Spring Boot, Thymeleaf, JWT, y JPA, con una visión empresarial sustentada en un análisis de mercado y un plan de desarrollo viable. A lo largo de este documento se explicará el proceso completo de diseño, implementación y validación de Kiricasa, así como sus funcionalidades, decisiones tecnológicas, estructura visual, y propuestas de mejora futuras.

Objetivos específicos

- Desarrollar un sistema de registro y autenticación de usuarios, integrando la verificación de correo electrónico y recuperación de contraseña.
- Implementar un sistema de roles que distinga entre usuarios normales, propietarios y administradores, con permisos diferenciados según el tipo de cuenta.
- Permitir la publicación de anuncios de alquiler, donde los usuarios puedan subir hasta 9 imágenes, introducir detalles como precio, barrio, número de habitaciones, tipo de vivienda, superficie, etc.
- Incorporar filtros avanzados de búsqueda, que permitan a los usuarios encontrar anuncios por criterios como barrio, tipo de habitación, número de compañeros, precio o fecha de publicación.
- Crear un sistema de favoritos y anuncios vistos, que permita a los usuarios guardar publicaciones de interés.
- Implementar un panel de administración para la gestión de publicaciones, usuarios y contenido, permitiendo a los administradores mantener la calidad del sitio.
- Desarrollar un blog o sección informativa, con artículos relacionados con la convivencia, el alquiler responsable y consejos útiles, con el objetivo adicional de mejorar el posicionamiento SEO.
- Garantizar la seguridad de la aplicación mediante autenticación con tokens JWT, cifrado de contraseñas con BCrypt, y control de acceso a rutas según el rol del usuario.

Tecnologías Utilizadas

Kiricasa ha sido planteado utilizando tecnologías modernas y ampliamente adoptadas en el desarrollo web, buscando garantizar la escalabilidad, seguridad y mantenibilidad del sistema. A continuación, se detallan las principales tecnologías utilizadas tanto en el **backend** como en el **frontend**, así como en la gestión del proyecto:

Backend

- **Spring Boot 3.2.x:** Framework que simplifica el desarrollo de aplicaciones Java basadas en Spring, facilitando la configuración automática, la gestión de dependencias y el despliegue rápido.
- **Spring Security:** Biblioteca utilizada para gestionar la autenticación y autorización en la plataforma, permitiendo el control de acceso a recursos según el rol del usuario.
- **JWT (JSON Web Tokens):** Sistema de autenticación sin estado que permite a los usuarios mantener la sesión activa mediante el envío de tokens en las peticiones HTTP.
- **JPA (Hibernate):** Capa de persistencia para el manejo de la base de datos mediante entidades Java, facilitando la interacción con MySQL de forma orientada a objetos.
- **MySQL:** Motor de base de datos relacional utilizado para almacenar toda la información de la plataforma: usuarios, publicaciones, imágenes, favoritos, etc.
- Heroku: Plataforma como servicio (PaaS) que permite desplegar, alojar y escalar aplicaciones en la nube de forma sencilla.
- Apache Tomcat: Servidor web y contenedor de servlets que actúa como entorno de ejecución para aplicaciones web Java.

Frontend

- **Thymeleaf:** Motor de plantillas que permite integrar HTML dinámico en las vistas del servidor, ideal para aplicaciones con renderizado en el backend como Kiricasa.
- **HTML5, CSS3 y JavaScript:** Tecnologías fundamentales para el desarrollo de las vistas, estilos y comportamiento del cliente.

Desarrollo de la aplicación

El proyecto ha sido todo un reto a la hora de desarrollo. Lo que más me costó entender fue la lógica que hay que seguir en Java en cuanto al backend, sobre todo porque venía de usar Laravel, donde el controlador, las rutas y la seguridad se gestionan de forma muy distinta. En Java, especialmente con Spring Boot, el enfoque es mucho más estructurado y modular. Ya que se incorpora una capa de “negocio” en la que hay que dividir mucho mejor las responsabilidades. No basta con tener un controlador y acceder a la base de datos directamente desde ahí, como se puede hacer en PHP: en Spring, cada clase tiene un rol específico. Al principio me costó entender esa separación, pero con el tiempo vi que permite tener un sistema más organizado y mantenable a largo plazo.

Además, tuve que acostumbrarme a usar `@RequiredArgsConstructor` para realizar la inyección de dependencias.

Siguiendo el desarrollo de la aplicación, opté por implementar un sistema de seguridad más moderno y robusto: **JWT (JSON Web Tokens)**. A diferencia del sistema tradicional de sesiones en el backend, JWT permite autenticar a los usuarios sin necesidad de mantener el estado en el servidor, lo cual es ideal para aplicaciones que podrían escalar.

En lugar de depender del control de sesión y proteger los formularios con tokens **CSRF** (Cross-Site Request Forgery), que es lo que suele usarse cuando se mantienen sesiones activas en el servidor, JWT se apoya en el **concepto de autenticación sin estado**.

En la clase SecurityConfig más concretamente en el método SecurityFilterChain es donde desactiva los tokens **CSRF**

```
    }
    .authenticationEntryPoint((request, response, authException) -> {
        response.sendError(HttpServletRequest.SC_UNAUTHORIZED, msg:"No autorizado")
    })
};

return http
    //desactiva la protección csrf
    ...
    .csrf(csrf -> csrf.disable())
    //desactiva la protección de cabeceras
    ...
    .authorizeHttpRequests(auth -> auth
        .dispatcherTypeMatchers(DispatcherType.FORWARD).permitAll()
        //rutas permitidas
        ...
        .requestMatchers(...patterns:"/", "/auth/**", "/nl/**", "/css/**",
            .anyRequest().authenticated()
        )
    )
}

sessionManagement(session -> session
```

Ilustración 1 Seguridad en la clase SecurityFilterChain

Siguiendo en esa clase es de las que más me costó ya que en ella se define toda la seguridad de la aplicación: qué rutas son públicas, cuáles requieren autenticación, cómo se gestiona la sesión, qué filtros se aplican y cómo se comporta el sistema al cerrar sesión o al denegar el acceso. Al principio me costó bastante porque no es simplemente "permitir o denegar" una ruta, sino que se combinan muchos elementos como los filtros personalizados, el manejo de errores, la política de sesiones y la configuración de logout.

Lo que más me confundía era cómo integrar correctamente mi filtro JWTAuthenticationFilter en el orden adecuado. Si no se colocaba antes del filtro UsernamePasswordAuthenticationFilter, el token no se procesaba correctamente y parecía que el usuario no estaba autenticado.

```

return http
    //desactiva la proteccion csrf
    .csrf(csrf -> csrf.disable())
    //desactiva la proteccion de cabeceras
    .authorizeHttpRequests(auth -> auth
        .dispatcherTypeMatchers(DispatcherType.FORWARD).permitAll()
        //rutas permitidas
        .requestMatchers(...patterns:"/","/auth/**","/nl/**", "/css/**", "/js/**", "/images/**","/uploads/**","variado/**").permitAll()
        .anyRequest().authenticated()
    )

    .sessionManagement(sess -> sess
        .sessionCreationPolicy(SessionCreationPolicy.IF_REQUIRED)
    )

    //este código personaliza la autenticación y el cierre de sesión en la aplicación, asegurando que se manejen correctamente las sesiones y los tokens JWT.
    .authenticationProvider(authenticationProvider)
    .addFilterBefore(jwtAuthenticationFilter, beforeFilter:UsernamePasswordAuthenticationFilter.class)
    .logout(logout -> logout
        .LogoutUrl(logoutUrl:"/logout")
        .LogoutSuccessUrl(logoutSuccessUrl:"/nl/home")
        .invalidateHttpSession(invalidateHttpSession:true)
        .deleteCookies(...cookieNamesToClear:"JSESSIONID")
    )
}

.build();

```

Ilustración 2 apartado donde se ve UsernamePasswordAuthenticationFilter

Otro apartado interesante ha sido ApplicationConfig, porque es donde se configura la lógica que se encargará de autenticar al usuario. Al principio puede parecer que solo es "rellenar beans", pero en realidad es clave para que todo funcione correctamente. Aquí definí cómo obtener un UserDetailsService personalizado, es decir, cómo cargar a los usuarios desde la base de datos usando mi UsuarioRepository.

También me costó entender cómo funciona el AuthenticationProvider y por qué se necesita uno. En mi caso usé DaoAuthenticationProvider para indicar que las contraseñas están encriptadas con BCrypt y que los datos del usuario vienen de la base de datos. Si algo no estaba bien configurado aquí, la autenticación fallaba aunque el resto del sistema pareciera correcto.

Es una clase que no se modifica mucho una vez está lista, pero sin ella el login no funcionaría correctamente y todo el sistema de seguridad se caería.

```

@Configuration
@RequiredArgsConstructor
/**
 * * * @author recur
 */
public class ApplicationConfig {
    private final UsuarioRepository ... usuarioRepository;

    @Bean
    public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws Exception {
        return config.getAuthenticationManager();
    }

    @Bean
    public AuthenticationProvider authenticationProvider () {
        DaoAuthenticationProvider provider = new DaoAuthenticationProvider();
        provider.setUserDetailsService(authenticationService());
        provider.setPasswordEncoder(passwordEncoder());
        return provider;
    }

    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Bean
    public UserDetailsService authenticationService() {
        return nombre -> ... .findByNameIgnoreCase(...).orElseThrow(() -> new UsernameNotFoundException(msg:"Usuario no encontrado"));
    }
}

```

Ilustración 3 ApplicationConfig

Dato ¿Qué es un @Bean?

Un bean es simplemente un objeto que es gestionado por el contenedor de Spring. Es decir, Spring se encarga de crear, inicializar y mantener ese objeto durante el ciclo de vida de la aplicación. Es una de las bases del desarrollo con Spring.

Por último, lo más destacable en cuanto a seguridad y filtro sería doFilterInternal de la clase JWTAuthenticationFilter por que corazón del sistema de autenticación con JWT. En este método se interceptan todas las peticiones y se comprueba si el usuario tiene un token válido antes de dejarle pasar. Es decir, aquí es donde realmente se decide si alguien puede acceder a una ruta o no.

Me costó bastante entender cómo se recupera el token, cómo se valida y cómo se le pasa la autenticación al contexto de Spring (`SecurityContextHolder`). También tuve que aprender a gestionar los casos especiales, como cuándo el token ha expirado, o si un usuario intenta acceder sin estar logueado.

```

        if (token == null) {
            Object sessionToken = request.getSession(create:false) != null
                ? request.getSession().getAttribute(name:"jwt")
                : null;
            if (sessionToken instanceof String) {
                token = (String) sessionToken;
            }
        }

        try {
            if (token != null && SecurityContextHolder.getContext().getAuthentication() == null) {
                String username = jwtService.getUsernameFromToken(token);
                if (username != null) {
                    UserDetailsService userDetailsService = userDetailsService.loadUserByUsername(username);
                    if (jwtService.isTokenValid(token, userDetailsService)) {
                        UsernamePasswordAuthenticationToken authToken =
                            new UsernamePasswordAuthenticationToken(userDetailsService, credentials:null, userDetailsService.getAuthorities());
                        authToken.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
                        SecurityContextHolder.getContext().setAuthentication(authToken);
                    }
                }
            }
        } catch (io.jsonwebtoken.ExpiredJwtException e) ...
        } catch (Exception e) {
            System.out.println("Token inválido o error: " + e.getMessage());
            request.getSession().removeAttribute(name:"jwt");
            request.getSession().invalidate();
        }

        if (isRutaPublica(path)) {
            filterChain.doFilter(request, response);
            return;
        }
    }
}

```

Ilustración 4 JWTAuthentificationFilter

Eso en cuanto a la seguridad para finalizar esta sección vamos a ver cómo es un controlador de Spring

```

    /**
 * Método para mostrar el formulario de edición de una publicación.
 * @param id
 * @param model
 * @param session
 * @return
 */
@GetMapping("/editar/{id}")
public String mostrarFormularioEdicion(@PathVariable("id") Long id, Model model, HttpSession session) {
    UsuarioModel usuario = (UsuarioModel) session.getAttribute(name:"usuario");
    if (usuario == null) return "redirect:/auth/login";

    PublicacionModel publicacion = publicacionRepository.findById(id).orElse(null);
    if (publicacion == null) return "redirect:/home";

    if (usuario.getRole().equals(UsuarioRol.ADMIN) && !publicacion.getUsuario().getId().equals(usuario.getId())) {
        return "redirect:/home";
    }
    List<String> fotos = new ArrayList<>(publicacion.getFotos());
    while (fotos.size() < 9) {
        fotos.add(e:"predeterminada.png");
    }
    model.addAttribute(attributeName:"fotosCompletas", fotos);

    model.addAttribute(attributeName:"usuario", usuario);
    model.addAttribute(attributeName:"publicacion", publicacion);
    model.addAttribute(attributeName:"barrios", barriosRepository.findAll());

    return "publicacion_editar";
}

```

Ilustración 5 controlador mostrar vista

En la imagen se puede ver un controlador que es el encargado de mostrar una vista.

Primero, se comprueba si el usuario está autenticado accediendo a su sesión. Si no lo está, se redirige al login. Luego, se busca la publicación con el ID recibido por la URL. Si no existe, se redirige al home.

Después, se añade una verificación importante: si el usuario no es administrador y la publicación no le pertenece, se le deniega el acceso. Esto evita que un usuario pueda modificar publicaciones de otros.

También se prepara la lista de imágenes para la galería de edición. Si hay menos de 9, se llenan los huecos con una imagen predeterminada para que el frontend tenga siempre la misma estructura.

Por último, se añaden los datos necesarios al modelo (usuario, publicación, barrios, fotos) y se retorna el nombre de la vista `publicacion_editar`.

Y para un POST

```

2   * Método para subir una imagen a una publicación.
3   *
4   * @param id           ID de la publicación.
5   * @param archivo       Archivo de imagen a subir.
6   * @param redirectAttributes Atributos para redirección.
7   * @return Redirección a La página de edición de La publicación.
8   */
9
10  @Param id
11  @Param archivo
12  @Param redirectAttributes
13  @Return
14  */
15
16  @PostMapping("/editar/{id}/subir-imagen")
17  @SuppressWarnings("CallToPrintStackTrace")
18  public String subirImagen(@PathVariable Long id,
19          @RequestParam("imagen") MultipartFile archivo,
20          RedirectAttributes redirectAttributes, Model model) {
21
22      PublicacionModel publicacion = publicacionRepository.findById(id).orElse(null);
23      if (publicacion == null) {
24          redirectAttributes.addFlashAttribute(attributeName:"error", attributeValue:"La publicación no existe.");
25          return "redirect:/home";
26      }
27
28      if (archivo == null || archivo.isEmpty()) {
29          redirectAttributes.addFlashAttribute(attributeName:"error", attributeValue:"No se seleccionó ninguna imagen.");
30          return "redirect:/detalle?id=" + id;
31      }
32
33      try {
34
35          String carpeta = publicacion.getCarpetasImagen();
36          if (carpeta == null || carpeta.isEmpty()) {
37              carpeta = "publicacion_" + publicacion.getId();
38              publicacion.setCarpetasImagen(carpetas);
39          }
40
41      }
42
43      catch (Exception e) {
44          redirectAttributes.addFlashAttribute(attributeName:"error", attributeValue:"Ocurrió un error al subir la imagen.");
45          return "redirect:/detalle?id=" + id;
46      }
47
48      return "redirect:/editar/" + id;
49  }

```

Ilustración 6 controlador post

```

public String subirImagen(@PathVariable Long id,
                        MultipartFile archivo) {
    File carpetaDir = new File(rutaBase, carpeta);
    if (!carpetaDir.exists()) carpetaDir.mkdirs();

    String nombreOriginal = archivo.getOriginalFilename();
    Path destino = Paths.get(carpetaDir.getAbsolutePath(), nombreOriginal);
    archivo.transferTo(destino.toFile());

    String rutaRelativa = nombreOriginal;

    for (int i = 0; i < 9; i++) {
        String imgActual = publicacion.getImagenPorIndice(i);
        if (imgActual == null || imgActual.isEmpty() || imgActual.equals("predeterminada.png")) {
            publicacion.setImagenPorIndice(i, rutaRelativa);
            break;
        }
    }

    publicacionRepository.save(publicacion);
    redirectAttributes.addFlashAttribute(attributeName:"success", attributeValue:"Imagen subida correctamente.");
} catch (IOException e) {
    e.printStackTrace();
    redirectAttributes.addFlashAttribute(attributeName:"error", attributeValue:"Error al guardar la imagen en disco.");
}

model.addAttribute(attributeName:"barrios", barriosRepository.findAll());
return "redirect:/detalle?id=" + id;
}
}
/** ...
 * @GetMapping("/editar/{id}/eliminar-imagen")
 * public String eliminarImagen(@PathVariable Long id,
 * ...
 */

```

Ilustración 7 controlador post

Este tema es igual que Laravel se recibe una petición con un post, hay que comprobar los campos tratarlos como se deseé guardar a la información y redirigir a una vista.

Planificación del proyecto

La planificación de Kircasa se ha estructurado en fases progresivas, permitiendo una implementación organizada y ajustada al calendario del curso. Cada etapa ha contado con objetivos concretos que han guiado el desarrollo del sistema.

Fases principales

- Análisis inicial y definición funcional

Se identificó la necesidad de una plataforma especializada en alquiler de habitaciones y pisos. Se establecieron los primeros requisitos funcionales y referencias visuales.

- Diseño del sistema

Se elaboró el diagrama de base de datos, guía de estilos, estructura de vistas y navegación general. Esto permitió definir una línea visual y técnica coherente desde el inicio.

- Desarrollo del backend y seguridad

Se implementó la lógica principal del servidor utilizando Spring Boot y Spring Security con JWT. Se desarrolló el sistema de roles, login, registro y recuperación de contraseña.

- Desarrollo del frontend y funcionalidades clave

Se crearon las vistas con Thymeleaf. Se integraron las funciones de publicar, filtrar, ver anuncios, añadir favoritos y gestionar el perfil del usuario.

- Pruebas, validaciones y documentación

Se realizaron pruebas funcionales para validar la lógica del sistema. Se corrigieron errores y se redactó la documentación técnica y la memoria del TFG.

Diagrama de Gantt

Para organizarme mejor, dividí el desarrollo del proyecto en semanas y tareas usando un diagrama de Gantt. Me ayudó a visualizar qué debía hacer en cada fase: análisis, diseño, backend, frontend, pruebas y despliegue. Aunque tuve que ajustar algunas cosas sobre la marcha, me sirvió para no perder el rumbo y llegar a tiempo con todo lo importante.

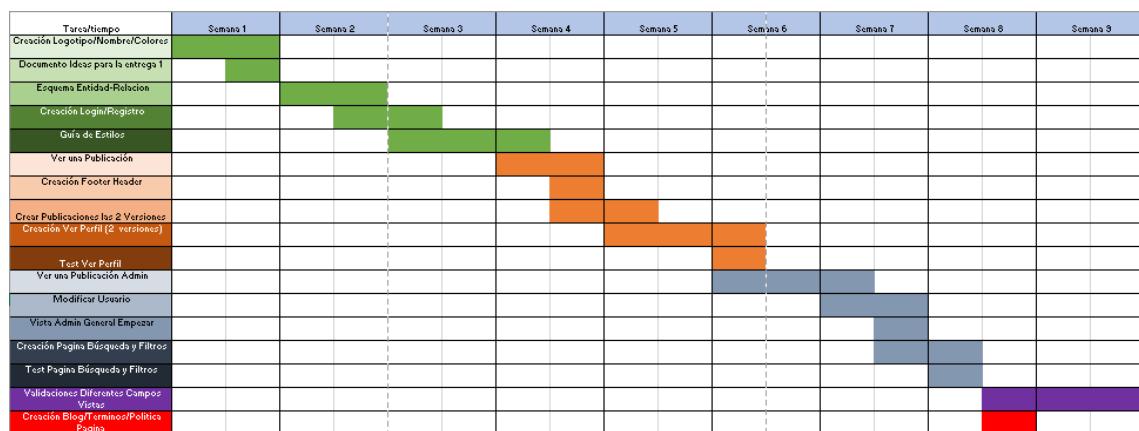


Ilustración 8 Diagrama de Gantt

Diagrama Entidad-Relación

El modelo entidad-relación define la estructura lógica de la base de datos de Kiricasa, reflejando los elementos clave del sistema y las relaciones entre ellos

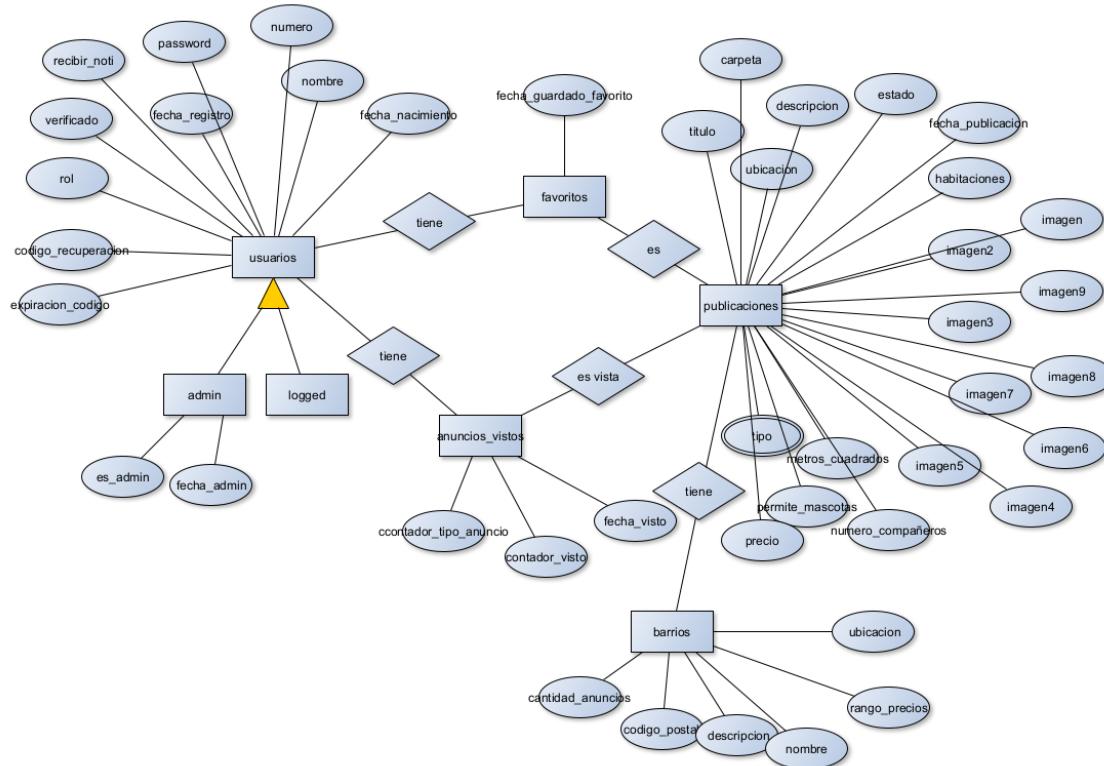


Ilustración 9 : Diagrama Entidad-Relación

1. Usuarios

Representa a todos los usuarios registrados en la plataforma, ya sean propietarios o administradores. Cada usuario posee los siguientes atributos:

- nombre, password, numero, fecha_nacimiento, correo, rol, fecha_registro
- Atributos relacionados con seguridad: verificado, recibir_noti, codigo_recuperacion, expiracion_codigo

Los usuarios pueden tener una relación con:

- admin (si el atributo es_admin está activo, y cuenta con fecha_admin)
- logged

Además, el usuario:

- Publica y visualiza anuncios
- Guarda favoritos
- Tiene anuncios vistos asociados, que a su vez incluyen contadores de tipo y ubicación

2. Publicaciones

Entidad central del sistema, representa los anuncios subidos por los usuarios. Atributos destacados:

- Información general: titulo, descripcion, tipo, precio, estado, fecha_publicacion, permite_mascotas
- Detalles del inmueble: ubicación, metros_cuadrados, numero_compañeros, habitaciones
- Sistema de imágenes: imagen1 hasta imagen9, organizadas mediante una carpeta

Una publicación:

- Pertenece a un barrio
- Es vista y es marcada como favorita por los usuarios

3. Barrios

Define los distintos barrios disponibles en la plataforma para facilitar los filtros de búsqueda. Atributos:

- nombre, ubicacion, codigo_postal, descripcion, rango_precios, cantidad_anuncios

Una publicación tiene un barrio asociado mediante una relación directa.

4. Favoritos

Tabla intermedia que enlaza a usuarios con publicaciones que han marcado como favoritas. Registra:

- fecha_guardado_favorito

Permite generar un histórico personalizado por usuario.

5. Anuncios Vistos

Entidad que refleja qué publicaciones ha visualizado un usuario. Atributos clave:

- fecha_visto, contador_tipo_anuncio, contador_visto

Es útil para implementar recomendaciones personalizadas y estadísticas.

Estructura del sitio web

La estructura de la página se ha pensado para tratar de hacer el acceso lo más fácil posible para el usuario acompañado de un diseño muy amigable y páginas muy intuitivas, vamos a hacer una recapitulación entre las diferentes versiones entre los distintos tipos de usuarios.

Cabecera (Header)

- Logotipo de Kiricasa, que redirige al inicio.
- Barra de búsqueda centrada, que permite filtrar por palabras clave.
- Lista de barrios en forma de botones, que permite filtrar rápidamente por zonas geográficas concretas de Zaragoza.
- Botón de acceso (solo visible para usuarios no logueados).
- Iconos de favoritos y perfil (visibles únicamente para usuarios logueados y admins).
- Botón de administrador acompañado del nombre del Admin (solo visible para administradores).

No logged



Ilustración 10 Header No Logged

En esta versión solo aparece el botón de iniciar sesión, la barra de navegación sigue funcionando, aunque no estés registrado.

Logged



Ilustración 11 : Header Logged

Aquí se puede observar que el color de la cabecera tiene una cierta variación demás de los botones que cambia por el del ver perfil y los favoritos del usuario.

Admin



Ilustración 12 : Header Admin

Aquí como se ha mencionado antes cambia la vista y tenemos cerrar sesión, ver perfil y la vista de administración.

Contamos en todas las versiones con un listado que se genera de forma dinámica con los barrios cada barrio que es añadirlo o borrado se genera de forma automática en la lista ya que se pasa como parámetros de a la vista, esto está realizado para que en un futuro si se desea ampliar el rango de la aplicación poder cambiar toda la página modificando menos contenido.

Homepage

Para esta hay muy pocas variaciones para los distintos usuarios solo cambia que a los usuarios loggeados y administradores les aparece un botón que les permite publicar anuncio y a los que no están registrados no. Además, que los filtros de barrio tanto como los que se especificaran a continuación filtran entre todos los anuncios según lo que el usuario clica por ejemplo si solo quiere ver pisos de Torrero-La Paz solo se le mostrarán pisos de esa ubicación.



Ilustración 13 : Diseño Principio HomePage

Y la versión de las personas que no están loggeadas

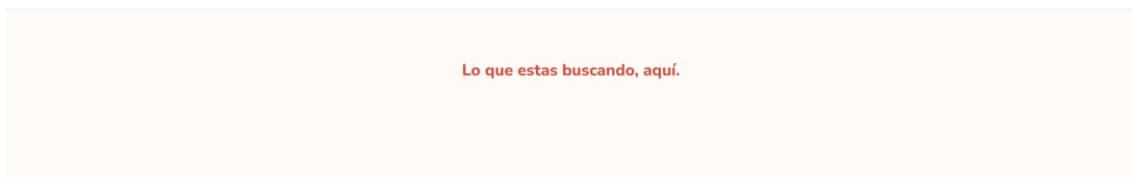


Ilustración 14 : Variacion HomePage No Logged

Seguimos con un grid que se genera de forma vertical con los anuncios y van apareciendo ciertos "desniveles" esto aumenta cuando hay más anuncios en la página para transmitir la sensación de movimiento de la página, y por último seguimos con un apartado que se diferencia por el tipo de piso.



Ilustración 15 : Filtros Generales HomePage

Footer

Para el footer es la misma estructura en todas las páginas y todos los usuarios, no cuenta con páginas de acceso restringidas por lo que no hay motivo para modificarlo.



Ilustración 16 : Diseño Footer

Perfil

Para el perfil de usuario habrá bastantes elementos, pero no hay variación como tal en esta vista de usuario a admin.

Para el administrador le parecerá las mismas opciones que a los usuarios

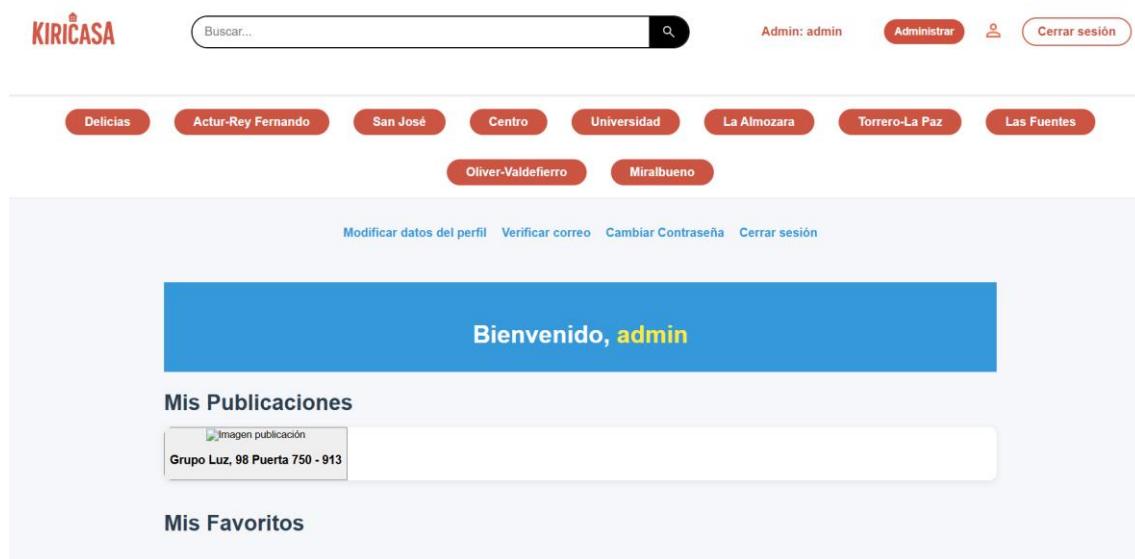


Ilustración 17 : Diseño Ver Perfil

Las fotos publicaciones se generarán de forma automática hacia la derecha al igual que los favoritos

A continuación, vamos a mostrar todas las opciones para que ofrece las opciones de usuario.

Editar Perfil

Nombre:

Email:

Número de teléfono:

Fecha de nacimiento:

Deseo recibir notificaciones por email

[Guardar cambios](#)
[Cancelar](#)

Ilustración 18: Diseño Editar Perfil

Para la edición de los datos de los datos de un usuario con varios apartados a cambiar

El siguiente apartado seria la verificación del correo que se enviara automáticamente un correo asociado a la cuenta con un código de verificación.

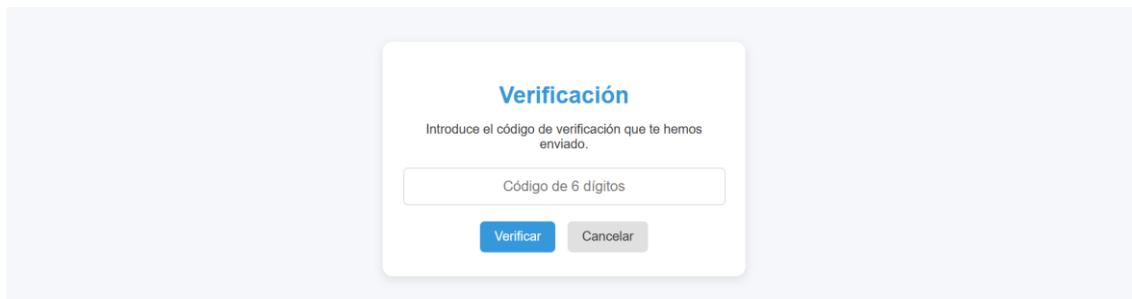


Ilustración 19: Vista Confirmación Correo



Ilustración 20: Vista Correo con Código

Tal que así una vez que esta verificado el perfil desaparecerá la opción del panel de usuario y en la BBDD quedará marcada como una cuenta verificada, esto estaba plateado para poder implementar ciertas funciones como para el solo puedan realizar ciertas acciones exclusivas para ellos.

Para el cambio de contraseña desde esta vista habrá que saberse su contraseña actual



The screenshot shows a modal dialog titled "Cambiar Contraseña". It contains three text input fields: "Contraseña actual:" (Current Password), "Nueva contraseña:" (New Password), and "Confirmar nueva contraseña:" (Confirm New Password). At the bottom are two buttons: "Guardar" (Save) in blue and "Cancelar" (Cancel) in grey.

Ilustración 21: Vista Cambiar Contraseña

Y se modificará con una confirmación.

Y por último un botón con la opción de cerrar sesión.

Publicaciones

Las imágenes se mostrarán como un carrusel de imágenes que ir cambiando según el usuario quiera.

Para las publicaciones hay 3 versiones ya que depende como se accede a ellas, para la versión del admin el podrá modificar cualquier publicación, así como eliminarla, en esta página el admin se ha pensado como un usuario más, pero con ciertos privilegios. Por lo

que se la vista sigue teniendo el apartado de favoritos y todo se muestra con normalidad.



Ilustración 22: Vista Ver Publicación Admin

El botón de añadir a favoritos cambiara según el estado del anuncio



Ilustración 24: Botón Quitar Favoritos

Ilustración 23 Botón Añadir Favoritos

Los pisos que uno mismo publica no se podrán guardar en favoritos y se podrán modificar y eliminar.

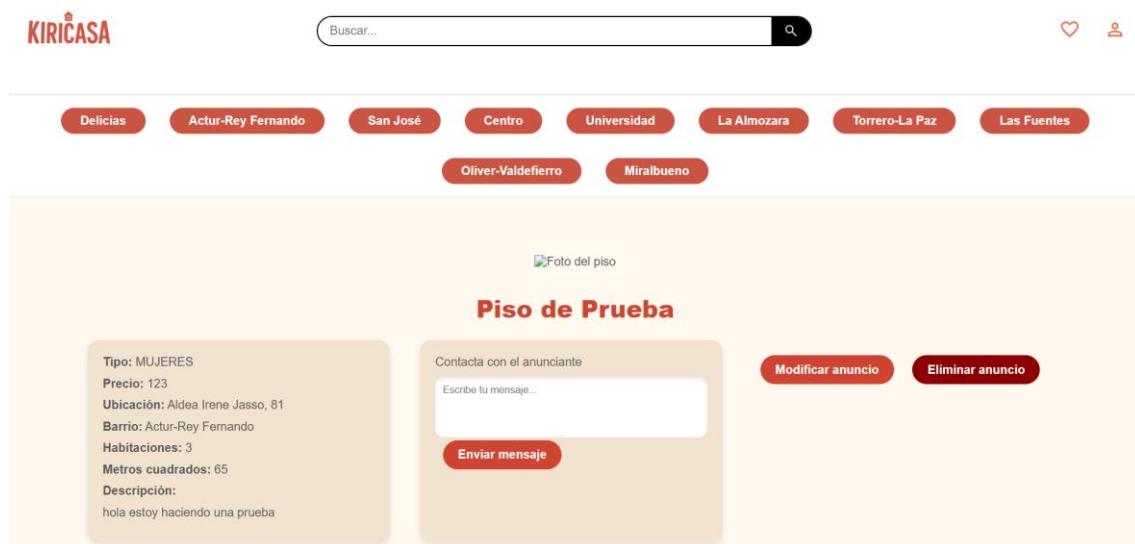


Ilustración 25 : Vista Publicación Subida Por Usuario

Para el apartado de modificar tenemos 2 secciones una con las fotos en la cual habrá 9 huecos en el que si una imagen falta se pondrá una imagen predeterminada y esta será clicarle y nos permitirá subir una foto.

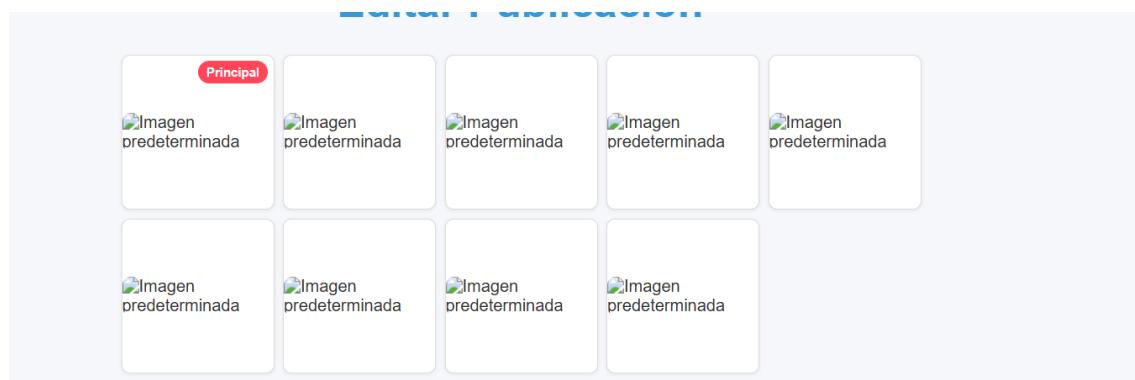


Ilustración 26: Vista Modificar Imagen

Cuando se quiera eliminar una foto saldrá una cruz en la imagen que esta subida que nos permitirá eliminara y la cual el hueco específico quedara libre además de que hay una imagen “principal” que saldrá de portada en el anuncio



Ilustración 27: Vista Modificar Imagen

Para la segunda sección saldrán toda la demás información del piso.

Título:	Piso de Prueba
Descripción:	hola estoy haciendo una prueba
Precio:	123
Estado:	Bien
Tipo de Piso:	MUJERES
Ubicación:	Aldea Irene Jasso, 81
Metros cuadrados:	

Ilustración 28: Vista Modificar Imagen

Con un botón de guardar cambios al final.

De este apartado solo queda comentar el apartado de mensajes que consta en que cuando nosotros metemos cualquier mensaje y le damos a enviar se nos abrirá sistema de correo electrónico que tengamos configurado en nuestro dispositivo y se rellenará solo con el cuerpo metido y el correo del usuario al que vamos a escribir.

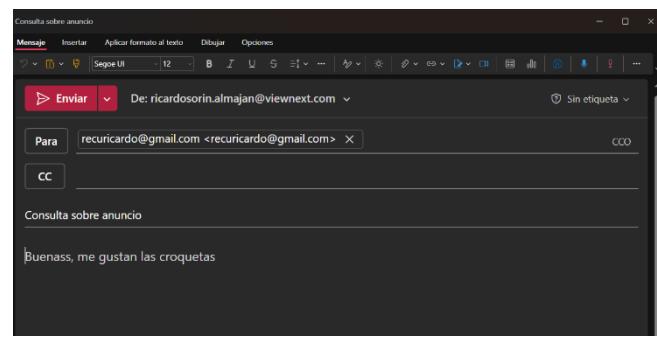


Ilustración 29: Vista del correo

Ilustración 30: Sección Enviar Mensaje

Subida de un Anuncio

Para la subida de un piso hay que rellenar todos los datos con sus respectivos datos

Publicar nuevo anuncio

Título:

Descripción:

Precio:

Estado:

Tipo de piso: HOMBRES

Ubicación:

Metros cuadrados:

Habitaciones:

Ilustración 31: Vista Publicar Anuncio

Habitaciones:

Permite mascotas

Número de compañeros:

Barrio:

▼

Subir imágenes (máx. 9):

Imagen 1	Imagen 2	Imagen 3
<input type="button" value="Seleccionar archivo"/> Ningún...cionado	<input type="button" value="Seleccionar archivo"/> Ningún...cionado	<input type="button" value="Seleccionar archivo"/> Ningún...cionado

Imagen 4	Imagen 5	Imagen 6
<input type="button" value="Seleccionar archivo"/> Ningún...cionado	<input type="button" value="Seleccionar archivo"/> Ningún...cionado	<input type="button" value="Seleccionar archivo"/> Ningún...cionado

Imagen 7	Imagen 8	Imagen 9
<input type="button" value="Seleccionar archivo"/> Ningún...cionado	<input type="button" value="Seleccionar archivo"/> Ningún...cionado	<input type="button" value="Seleccionar archivo"/> Ningún...cionado

Ilustración 32: Vista Publicar Anuncio

Como se puede observar las fotos vienen especificadas con la posición que ocuparan en todo momento.

Búsqueda

KIRICASA

Buscar...

Resultados para:

Título Ubicación Estado Precio máx.
 Min. m² Habitaciones Compañeros Permite mascotas

Prado, Maestas y Jaimes Asociados

Consequatur est consequatur laboriosam dolor. Dolores exercitationem voluptas rem. Iure unde dolore est rem hic illo. Cum a labore velit blanditiis adipisci.

Ilustración 33: Vista Búsqueda

La búsqueda filtra por título tipo y ubicación, pero si no estamos conformes a la hora de la búsqueda nos aparece una opción con todos los filtros. Se puede buscar sin ninguna palabra y nos saldrán todos los anuncios.

Paginas Varias

En este apartado tenemos las 3 páginas de footer en primer lugar tenemos el:

Blog

Blog de Kircasa

Consejos para encontrar piso en Zaragoza

Publicado el 15 de mayo de 2025

Buscar piso en Zaragoza puede ser todo un reto si no sabes por dónde empezar. La ciudad cuenta con una gran variedad de barrios, cada uno con sus propias ventajas. En Kircasa recomendamos comenzar por zonas como **Actur** (ideal para familias y estudiantes del Campus Río Ebro), **Universidad** (ambiente joven y buena conexión de transporte) o **La Almozara** (zona tranquila con buenos precios).

Establece un presupuesto mensual que incluya alquiler, suministros, transporte y gastos personales. Utiliza filtros por barrio, precio y características clave como si acepta mascotas o si está amueblado. No dudes en contactar al propietario para resolver dudas, y si es posible, agenda una visita presencial antes de tomar una decisión.

¿Qué tener en cuenta al compartir piso?

Publicado el 7 de mayo de 2025

Ilustración 34: Vista contenido Blog

Política de Privacidad

Política de Privacidad

1. Información general

En Kiricasa, la privacidad de nuestros usuarios es una prioridad. Esta Política de Privacidad describe cómo recopilamos, usamos y protegemos tu información personal cuando utilizas nuestra plataforma.

2. Datos que recopilamos

Cuando te registras en Kiricasa o utilizas nuestros servicios, podemos recopilar la siguiente información:

- Nombre completo
- Dirección de correo electrónico
- Teléfono (opcional)
- Preferencias de búsqueda y anuncios vistos
- Direcciones IP, dispositivo y datos de navegación

3. Uso de los datos

Utilizamos tus datos para los siguientes fines:

Ilustración 35 : Vista contenido Privacidad

Términos y Condiciones de Uso

Términos y Condiciones de Uso

1. Aceptación de los términos

Al acceder y utilizar la plataforma Kiricasa, aceptas cumplir con estos Términos y Condiciones. Si no estás de acuerdo con alguno de ellos, te recomendamos no utilizar nuestros servicios.

2. Uso de la plataforma

Kiricasa es un servicio destinado a facilitar la publicación, búsqueda y contacto de anuncios inmobiliarios. Está prohibido el uso de la plataforma para fines fraudulentos, ilegales o que atenten contra los derechos de terceros.

3. Registro y cuentas de usuario

Para acceder a ciertas funcionalidades, el usuario debe registrarse. El usuario se compromete a proporcionar información veraz, mantener la confidencialidad de sus credenciales y notificar cualquier uso no autorizado de su cuenta.

4. Publicación de anuncios

Los usuarios son responsables del contenido que publican. Kiricasa se reserva el derecho a editar, ocultar o eliminar anuncios que incumplan nuestras políticas, contengan datos falsos, ofensivos o inapropiados.

Ilustración 36 : Vista contenido Términos

Administración

Esta vista es la exclusiva de los administradores, y en ella se encuentran varios apartados

Panel de Administración

Gestión de Usuarios

ID	Nombre	Email	Rol	Acciones	
1	admin	admin@kiricasa.com	ADMIN	<button>Borrar</button>	<button>Hacer Admin</button>
2	lourdes.nazario	luis.mares@hotmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>
3	laura.menchaca	ignacio.galvan@hotmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>
4	juana.sisneros	juancarlos.concepcion@gmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>
5	antonio.pulido	sergio.altamirano@hotmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>
6	cristina.vázquez	lorena.hernandez@hotmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>
7	rubén.mojica	rafael.caban@hotmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>
8	maríaelena.reyes	antonia.villasenor@hotmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>
9	ramón ortega	caridad.muñoz@gmail.com	USER	<button>Borrar</button>	<button>Hacer Admin</button>

Ilustración 37 : Vista Panel Admin

El primero que se observa es una lista con todos que no permitirá borrarse a uno mismo ni quietarse el rol de administrador

Añadir Nuevo Barrio

Nombre del barrio:	<input type="text"/>
Descripción:	<input type="text"/>
Ubicación:	<input type="text"/>

Lista de Barrios

ID	Nombre	Descripción	Ubicación	Acciones
1	Delicias	Barrio popular y multicultural con mucha vida comercial	Avenida de Madrid	<button>Borrar</button>
2	Actur-Rey Fernando	Moderno y bien comunicado, junto al río Ebro	Calle Gertrudis Gómez de Avellaneda	<button>Borrar</button>
3	San José	Zona tradicional con buen acceso al centro y servicios	Avenida San José	<button>Borrar</button>
4	Centro	Plaza mayor y corazón económico de la ciudad	Paseo de la Alameda	<button>Borrar</button>

Ilustración 38 : Vista Panel Admin

Seguimos con la opción de dar de alta un nuevo barrio y después una lista con todos los cuales nos permitirá eliminarlos.

Plan de Empresa

1.-Oportunidades de negocio y necesidades demandadas.

La aplicación propuesta responde a una necesidad real y creciente: la búsqueda de pisos y habitaciones en alquiler, especialmente en grandes ciudades o zonas con alta demanda de vivienda como Zaragoza. Plataformas como Wallapop o Milanuncios ya explotan este nicho, pero el autor busca diferenciarse mediante funcionalidades específicas y una interfaz más limpia e intuitiva. La posibilidad de filtrar por número de compañeros, tipo de habitación, precio o fecha de publicación apunta a cubrir de manera más precisa las necesidades de los usuarios.

Justificación de la oportunidad de negocio

El modelo de negocio se justifica por la alta demanda de alquiler compartido, especialmente entre estudiantes, jóvenes profesionales y personas con movilidad geográfica frecuente. Aunque existen plataformas consolidadas, el enfoque está en mejorar la experiencia de usuario, simplificar el proceso de publicación y búsqueda, e incorporar funciones sociales como el chat o el blog para fomentar el engagement. Además, se propone un panel de administración, para una futura escalabilidad del proyecto.

Descripción de la localización de la empresa.

La primera idea es orientarla para Zaragoza, pero al ser un modelo se le podrán cambiar los datos por si en un futuro se desea expandir a otras ciudades siguiendo la misma lógica-

Estudio de nuestro cliente y segmentación de mercado

- Usuarios jóvenes (estudiantes o trabajadores de entre 18 y 35 años).
- Personas interesadas en alquiler de habitaciones o pisos compartidos.
- Residentes actuales o futuros de Zaragoza.
- Personas con conocimientos básicos de navegación web.

2. Relación de productos o servicios a prestar. Comenta los aspectos destacados acerca de estos apartados.

Mi proyecto consiste en una plataforma web de anuncios de alquiler de pisos y habitaciones, orientada principalmente a usuarios que buscan o publican alojamiento en Zaragoza. El objetivo es facilitar el encuentro entre oferta y demanda, mejorando la experiencia frente a otras plataformas más genéricas.

2.1 Producto/servicio

El servicio principal que ofrezco es una web intuitiva y especializada donde los usuarios pueden:

- Publicar anuncios de habitaciones o pisos en alquiler.
- Buscar alojamiento mediante filtros específicos (tipo de vivienda, barrio, número de compañeros, precio, etc.).
- Comunicarse entre ellos mediante un sistema de mensajería interna (o por correo si no se implementa el chat).
- Gestionar sus publicaciones y su perfil.
- Acceder a un blog con contenido útil sobre vivienda, convivencia y consejos de alquiler, enfocado al posicionamiento SEO.

Mi idea es diferenciarme con una interfaz limpia, rápida y centrada solo en Zaragoza, evitando la sobrecarga de anuncios irrelevantes que se ve en otras plataformas.

2.2 Precio

Inicialmente, el servicio será gratuito tanto para usuarios que publican como para los que buscan, con el fin de atraer tráfico y crecer en popularidad. A medio plazo, tengo pensado implementar una opción de anuncios destacados o premium con una pequeña tarifa para quienes quieran que su anuncio aparezca primero o con más visibilidad.

Esta estrategia “freemium” me permitirá mantener el acceso libre y popularizar el sitio, pero a la vez empezar a monetizarlo de forma progresiva.

2.3 Promoción

Para promocionar la plataforma tengo previsto usar varias estrategias:

- Redes sociales, sobre todo Instagram y TikTok, donde los jóvenes de Zaragoza pasan mucho tiempo.
- Publicidad en grupos de estudiantes y alquiler en Telegram, Facebook y foros locales.
- SEO local, aprovechando el blog para posicionar palabras clave como “alquiler habitaciones Zaragoza” o “buscar piso Zaragoza”.
- A medio plazo, me gustaría colaborar con residencias, universidades y academias de Zaragoza para dar a conocer la página entre nuevos estudiantes o personas que se trasladan a vivir a la ciudad.



Ilustración 39 : Ilustración Redes

2.4 Distribución

El servicio se prestará completamente online a través de mi página web. No necesito punto físico, ya que todo el proceso se realiza de forma digital: publicar, buscar y contactar. Además, al estar enfocado solo en Zaragoza, puedo ofrecer recomendaciones y contenidos locales más ajustados a lo que los usuarios necesitan.

La idea es que cualquier persona con acceso a internet pueda usar la plataforma desde su móvil, tablet o PC. Quiero que sea ligera, responsive y muy accesible desde cualquier dispositivo.

3. Constitución de la empresa

Para poner en marcha este proyecto de plataforma web de alquileres centrada en Zaragoza, he considerado la forma jurídica más adecuada en función de los recursos disponibles, la responsabilidad legal y la facilidad de gestión.

3.1. Tipo de sociedad elegido explicando las razones

He optado por constituir una Sociedad Limitada (SL). Las razones de esta elección son:

- Responsabilidad limitada al capital aportado, lo cual protege mi patrimonio personal en caso de deudas o problemas legales.
- Es una forma muy común para pequeños proyectos y emprendedores individuales, como el mío, ya que permite empezar con un capital social mínimo de 1 euro (aunque lo ideal es aportar algo más).
- Me permite crecer en el futuro, incorporar socios o colaboradores fácilmente y tener una estructura más profesional de cara a posibles acuerdos o inversiones.
- A nivel fiscal, aunque tiene más obligaciones que un autónomo, también puede ofrecer ventajas a medio plazo si el proyecto genera ingresos sostenidos.



Ilustración 40 : Ilustración libros

3.2. Nombra la relación de trámites mercantiles, fiscales, laborales y locales que debe realizar la empresa

Para constituir la empresa de forma legal, debo realizar los siguientes trámites:

Mercantiles:

- Certificación negativa del nombre en el Registro Mercantil Central.
- Redacción de los estatutos y firma de la escritura de constitución ante notario.
- Inscripción de la empresa en el Registro Mercantil de Zaragoza.

Fiscales:

- Darse de alta en el censo de empresarios (modelo 036).
- Elegir el régimen de IVA e impuesto de sociedades correspondiente.

Laborales (si contrato trabajadores):

- Alta de la empresa en la Seguridad Social.

Locales:

- Al ser una actividad completamente online y sin atención presencial, no necesito licencia de apertura física, pero si llegara a abrir una oficina en Zaragoza, tendría que tramitar la correspondiente licencia municipal en el Ayuntamiento.
- También debo asegurarme de cumplir con la Ley de Protección de Datos (LOPD), ya que trataré datos personales de los usuarios, lo cual implica registrar los ficheros y disponer de políticas de privacidad adecuadas.

4. Recursos

Para llevar a cabo este proyecto necesito una combinación de recursos técnicos, humanos y económicos que me permitan desarrollar, lanzar y mantener la plataforma en condiciones óptimas.

4.1.-Recursos Materiales/Técnicos

Dado que se trata de un proyecto digital, los recursos materiales no son muchos, pero sí esenciales:

- Ordenador personal con buen rendimiento para programar, hacer pruebas y gestionar la web.
- Conexión a Internet estable y rápida, fundamental para el desarrollo y gestión de la plataforma.
- Servidor web (puede ser compartido en la fase inicial o VPS si el tráfico aumenta).
- Dominio y certificado SSL para seguridad y presencia profesional (ej. www.alquilazaragoza.com).

- Software de desarrollo web: editores de código (como Visual Studio Code), bases de datos (MySQL o similar), herramientas de control de versiones (Git).
- Framework de desarrollo (Laravel, Django, etc.), y recursos adicionales como bibliotecas, APIs o servicios externos si se requieren (por ejemplo, para mapas o correo electrónico).

4.2.-Recursos Humanos

Al tratarse de un proyecto que comienzo en solitario, en esta fase inicial yo mismo asumo todas las funciones: desarrollo, diseño, gestión y promoción.

A medida que el proyecto crezca, tengo previsto incorporar o colaborar con:

- Un diseñador gráfico o UX/UI para mejorar la experiencia del usuario.
- Un especialista en marketing digital para campañas SEO/SEM y redes sociales.
- Un técnico en mantenimiento de servidores, si se hace necesario un entorno más profesional y estable.

También podría recurrir a freelancers puntuales para tareas específicas como ilustraciones, redacción de contenido o revisión legal de los textos de la web.

4.3.-Recursos Económicos

El presupuesto inicial del proyecto es ajustado, pero suficiente para cubrir los elementos básicos. Estimo una inversión inicial baja porque:

- Desarrollo y programación los realizo yo mismo.
- Utilizaré un servidor económico en fase de lanzamiento.
- Los costes principales serán:
 - Dominio web y servidor: ~100 €/año.
 - Software y herramientas: uso de herramientas gratuitas o de bajo coste.
 - Promoción en redes sociales: inversión inicial de ~200 €.
 - Alta de la empresa (notaría, registro, etc.): alrededor de 300–600 €.

Para financiarlo, usaré recursos propios y ayudas para emprendedores si puedo optar a alguna a nivel local (como las del Ayuntamiento de Zaragoza o programas para jóvenes emprendedores del INAEM).

Conclusiones

Con este proyecto quiero ofrecer una solución útil y especializada para quienes buscan o publican pisos y habitaciones en Zaragoza, mejorando la experiencia actual que ofrecen plataformas más genéricas. A través de un diseño claro, funciones prácticas y una orientación local, puedo cubrir una necesidad real del mercado, especialmente entre jóvenes y estudiantes.

He optado por una Sociedad Limitada para proteger mi responsabilidad personal y permitir el crecimiento futuro del proyecto. La inversión inicial es asumible, ya que puedo desarrollar gran parte por mi cuenta, y planeo una monetización progresiva mediante servicios premium.

Los recursos materiales y humanos necesarios son razonables y están bien definidos, yuento con una estrategia de promoción realista basada en redes sociales, SEO y colaboraciones locales.

En resumen, considero que el proyecto es viable, escalable y bien enfocado, con posibilidades reales de consolidarse si logro una buena base de usuarios en Zaragoza desde el inicio.

Guía de Estilos

Introducción

En este documento se definirán los aspectos visuales de la página web de Kiricasa, pasando por el layout, elementos visuales, colores e imágenes, para poder hacer que el desarrollo de las vistas de esta página sea más ameno y más ordenado. Se definirá el apartado visual de todas las vistas independientemente de su nivel, pero antes de llegar a eso primero hay que definir los colores que va a tener la página.

Por lo que lo primero que hay que tener en cuenta es el logo de la página, que en este es bastante sencillo intentando representar al máximo la página con un diseño relajado pero firme.

Logotipo

El logotipo está compuesto del nombre de la página con una casa como elemento decorativo, el color de la letra es #cc5443

Y esta con la fuente de texto de sans serif, con formas redondeadas y un diseño amigable.

Algunas de las fuentes parecidas a esta que se usaran en la página son:

Baloo, Nunito ExtraBold, Cabin Rounded

Fuentes

sans serif

Títulos Principales

Transmite una estética moderna, clara y profesional. Este tipo de letra, al carecer de remates, facilita una lectura rápida y directa, lo cual es ideal para captar la atención del usuario desde el primer vistazo. Además, su estilo limpio y minimalista armoniza con las tendencias actuales de diseño web, mejorando la jerarquía visual y reforzando la identidad digital del sitio.

Nunito ExtraBold

Secciones

Para el contenido de las secciones se ha elegido la tipografía Nunito ExtraBold por su excelente legibilidad y su presencia visual sólida, que permite destacar los bloques de información de forma clara y atractiva. Su diseño redondeado combina suavidad con fuerza visual, lo que aporta un equilibrio entre profesionalismo y accesibilidad.

Baloo

Títulos Secundarios

la tipografía Baloo debido a su estilo amigable, redondeado y ligeramente informal, que aporta cercanía y calidez al diseño sin perder legibilidad. Esta elección contrasta de forma equilibrada con la seriedad y limpieza de la tipografía sans serif utilizada en los títulos principales, creando una jerarquía visual clara y atractiva.

Cabin Rounded

Cuerpo de texto

Para el cuerpo de texto se ha seleccionado la tipografía Cabin Rounded por su apariencia limpia, moderna y ligeramente suavizada, que facilita una lectura cómoda y fluida en párrafos extensos. Sus terminaciones redondeadas aportan un tono amable y accesible, lo que mejora la experiencia del usuario al recorrer el contenido principal.

Cabecera

Como se observa en la imagen y siguiendo las instrucciones del documento del análisis de la aplicación, las medidas son las indicadas en la imagen, están representadas en porcentajes para que se pueda realizar su diseño responsive como se comentó, tendremos 2 headers en la imagen se puede ver la cabecera de los usuarios que no están registrados

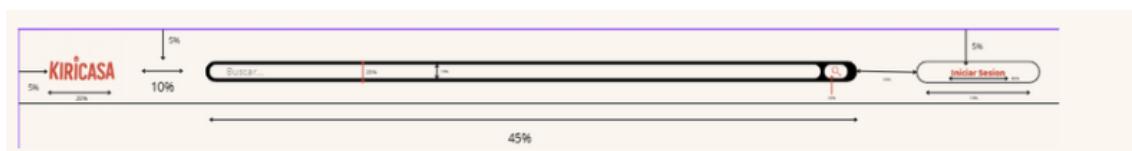


Ilustración 41 : Medidas Cabecera No Logged

Por otro lado, en este apartado se puede observar la cabecera de los usuarios que si están registrados teniendo los dos botones más igual que se mencionó en el documento anterior

Layout de la homepage

El diseño de la página como está definido está basado en Wallapop con un diseño más informal pero agradable siguiendo un orden pero dando la sensación de caos tanto color la búsqueda como el título deberán de estar centrados en la página teniendo una medida lateral de 35% y 60% respectivamente con un margen entre ellos de 3% y una altura de 20% y 10%, teniendo un margen el conjunto de objetos de 10% con todos los elementos que le rodean, los elementos donde irán los pisos quería que se generen aleatoriamente en un espacio marcado, establecían un patrón aleatorio, además todos los elemento tendrán un 5% de margen con el otro en todas las direcciones y el tamaño será completamente aleatorio siendo la máxima cantidad en el eje horizontal de 5 elementos y el vertical de 8

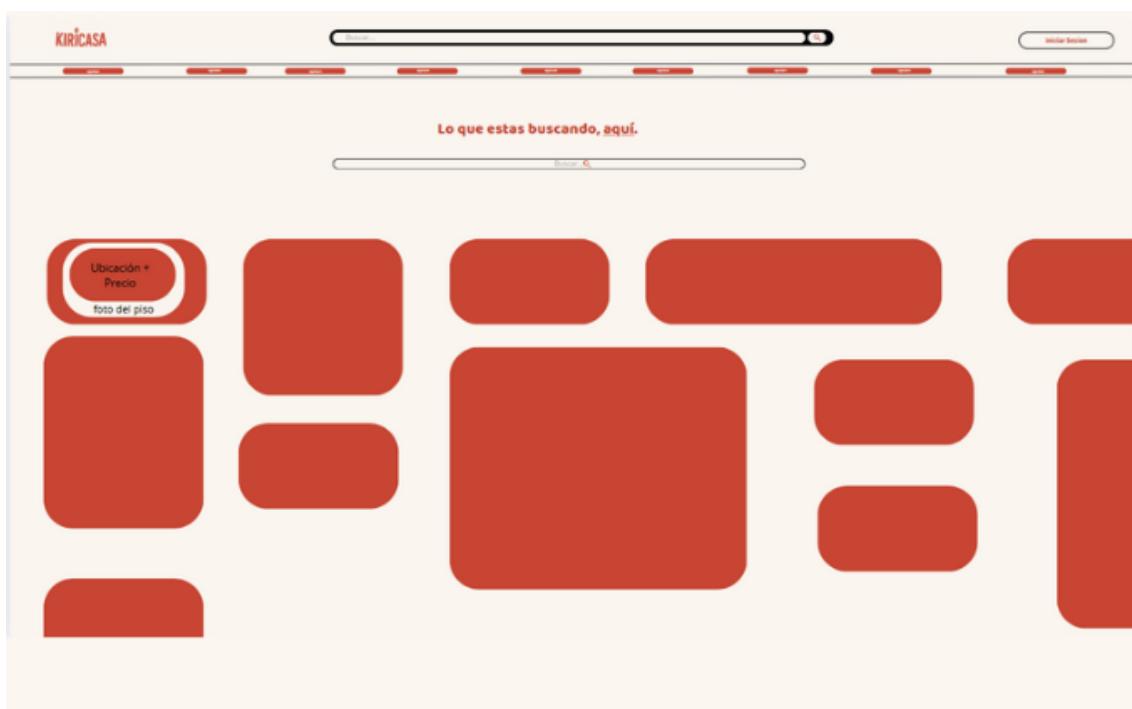


Ilustración 42 : Layout de la homepage

Aquí se puede ver definido con exactitud, como lo he dicho anteriormente, los elementos tendrán diferentes tamaños por lo que se mantendrá esta estructura de ellos

siendo la misma en todos los casos, siendo las fotos del piso las subidas por el usuario pasando previamente por un “control” para asegurarse que la calidad sea aceptable

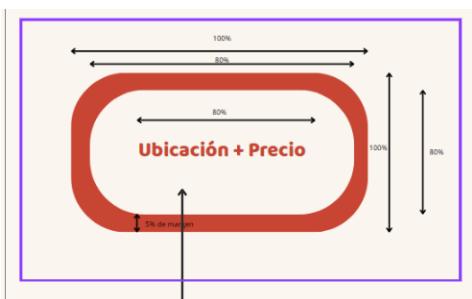


Ilustración 43 : Diseño Sección HomePage

Aquí se puede observar la continuación de la página, ya que quiero añadir una pequeña sección en la cual se mostrarán unas categorías con los pisos. Siguiendo el ejemplo de Wallapop y Milanuncios, las categorías serán los distintos barrios de Zaragoza y los que haya registrados en la base. Se escogerán 4 al azar de los que haya y se mostrarán las categorías de ellos. **En cuanto a las medidas, cada categoría contará con una altura de unos 25 cm y una anchura de 15 cm, con una separación del primer elemento y el último de 10% y una separación de cada elemento de 15%. Contará con una separación de arriba y abajo de 10%.**



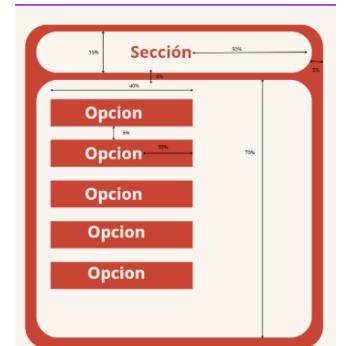
Ilustración 44 : Diseño Final HomePage

Diseño del footer

El sueño de cada sección será siguiendo este patrón, y en cada sección habrá diferentes opciones. La primera será con el nombre de la página con la posibilidad de poder acceder

a distintas opciones de la misma, la segunda con un aviso legal el cual tendrá varias opciones y demás...

Para las medidas se siguen las indicadas en la imagen, ya que en cada sección tendrá las mismas medidas, salvo la de la segunda fila, la cual tendrá una altura de 15 cm y una anchura equivalente a los 5 elementos superiores.



Login

Como se mencionó el apartado de login/registro será una ventana superpuesta a la página en la que esté el usuario. Esta ventana facilitará el acceso a la plataforma de manera rápida y eficiente, sin interrumpir la experiencia de navegación del usuario. Al abrirse, la ventana mostrará un diseño limpio y sencillo, con campos claros para ingresar el nombre de usuario y la contraseña, junto con opciones para recuperar la contraseña o registrarse si aún no se tiene una cuenta.

El cuadro que aparece tendrá un margen de un 30% superior e inferior y un 25% lateral, contará con una altura de 40% y una anchura de 50%. Después estará separado por 4 filas, esas tendrán un padding de un 5% y todas estarán centradas. El texto y el botón tendrán 2 em de separación, y cada elemento estará separado por 4 em. Por último, los botones tendrán que tener una separación de 2 em.

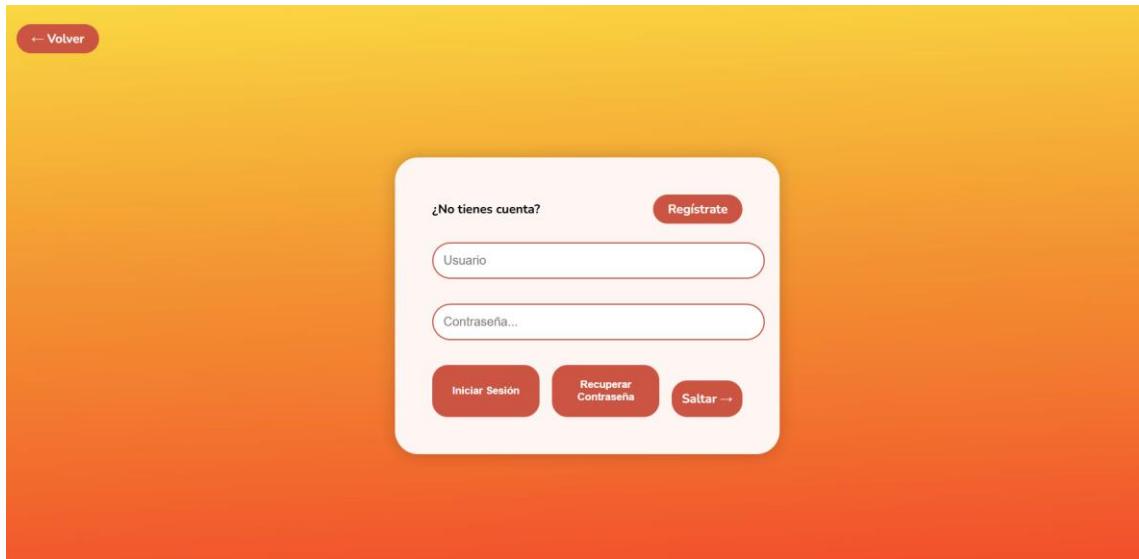


Ilustración 46 Vista Login

Registro

La estructura es similar a la pestaña de inicio de sesión, solo que aparecen varios campos necesarios para el registro de la página. Para acceder, será necesario aceptar los términos y condiciones de la página.

Para asegurar que la página web sea adaptable a diferentes tamaños de pantalla, se recomienda que el contenedor principal ocupe entre el 80% y 90% del ancho disponible, centrado mediante margin: auto. Los formularios o tarjetas de registro pueden tener un ancho entre 40% y 60%, dependiendo del diseño general y del dispositivo. Los inputs de texto deberían ocupar el 100% del ancho del formulario para mantener uniformidad. Además, los botones pueden establecerse en un 30% a 50% del ancho del formulario, garantizando que sean visibles, pero sin dominar el espacio.

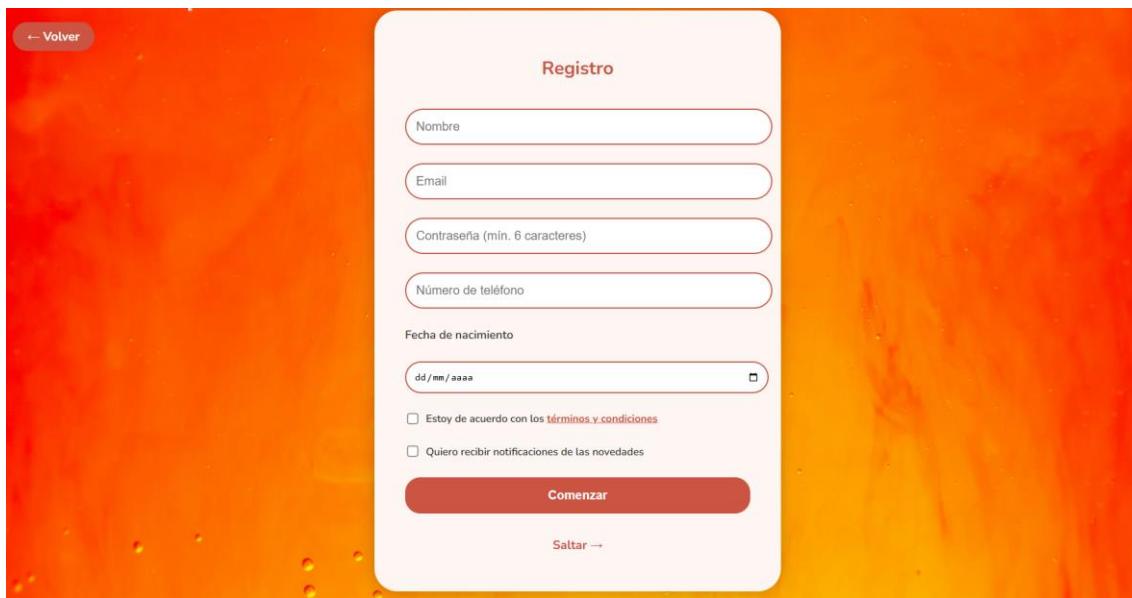


Ilustración 47 : vista Registro

Contraseña olvidada

Para mantener una estructura coherente en la página de recuperación de contraseña, se recomienda que el contenedor principal tenga un ancho de entre 30% y 50% respecto al total de la pantalla, y esté centrado con margin: auto. Los campos de entrada (input) deben ocupar el 90% a 100% del ancho del formulario para asegurar legibilidad y facilidad de uso. El botón de acción debería tener un ancho de entre 30% y 40% del contenedor, manteniendo un tamaño accesible pero proporcionado dentro del diseño general.



Ilustración 48 : vista Recuperar Contraseña

Visualización de un anuncio

Para estructurar la página de detalle del piso, se recomienda que el carrusel de imágenes ocupe entre el 80% y 90% del ancho total y esté centrado horizontalmente. La sección de título puede abarcar el 80% del ancho de la pantalla. En la sección inferior, el bloque de información del piso debería ocupar aproximadamente el 65% del ancho, y el bloque de contacto el 30%, dejando un pequeño margen entre ambos. Los botones como "Enviar mensaje" pueden configurarse para usar entre 40% y 60% del ancho de su contenedor para mantener un equilibrio visual y buena usabilidad.



Ilustración 49 : Vista publicación usuario loggeado

[Ver perfil](#)

El diseño de la página se organiza con un encabezado superior que ocupa aproximadamente el 10% de la altura total, en el que el buscador tiene un ancho del 40% y el botón “Iniciar sesión” un 7-10%. Justo debajo, la sección principal se distribuye en tres columnas: a la izquierda, la foto de perfil ocupa un 15% del ancho y un 30% de la altura; al centro, la información del usuario abarca cerca del 45% del ancho con la misma altura; y a la derecha, tres botones (Activar 2FA, Modificar datos y Borrar cuenta) con un ancho de 12-15% y una altura de 6-8% cada uno, separados entre sí por un 2-3% de espacio vertical. En la zona inferior se encuentra una fila de cuatro bloques: el primero, destinado a “Anuncios”, ocupa un 18-20% del ancho y entre un 20-25% de la altura, seguido de tres bloques iguales que miden un 18% de ancho cada uno, con separación horizontal del 2-3%. Finalmente, en la parte más baja, hay una barra de navegación o progreso que tiene un ancho del 20% y una altura de aproximadamente 3-4%.



Ilustración 50 : vista panel de ver perfil

Modificar perfil

El diseño de la página se organiza con un encabezado superior que ocupa aproximadamente el 10% de la altura total, en el que el buscador tiene un ancho del 40% y el botón “Iniciar sesión” un 7-10%. Justo debajo, la sección principal se distribuye en tres columnas: a la izquierda, la foto de perfil ocupa un 15% del ancho y un 30% de la altura; al centro, la información del usuario abarca cerca del 45% del ancho con la misma altura; y a la derecha, tres botones (Activar 2FA, Modificar datos y Borrar cuenta) con un ancho de 12-15% y una altura de 6-8% cada uno, separados entre sí por un 2-3% de espacio vertical. En la zona inferior se encuentra una fila de cuatro bloques: el primero, destinado a “Anuncios”, ocupa un 18-20% del ancho y entre un 20-25% de la altura, seguido de tres bloques iguales que miden un 18% de ancho cada uno, con separación horizontal del 2-3%. Finalmente, en la parte más baja, hay una barra de navegación o progreso que tiene un ancho del 20% y una altura de aproximadamente 3-4%.

Editar Perfil

Nombre:
ric11

Email:
recunardo@gmail.com

Número de teléfono:
6437145

Fecha de nacimiento:
dd/mm/aaaa

Deseo recibir notificaciones por email

Guardar cambios Cancelar

Ilustración 51 : Vista de editar perfil

Activación verificación 2 pasos

El diseño de esta pantalla está enfocado completamente en la funcionalidad de confirmar el correo electrónico del usuario. En el centro de la página, hay un cuadro blanco de esquinas redondeadas que ocupa cerca del 30-35% del ancho de la pantalla y alrededor del 25% de la altura. Este cuadro aparece sobre un fondo uniforme en tono salmón rojizo, generando un contraste agradable a la vista. En la parte superior del cuadro, se encuentra el título “Confirmación de correo electrónico”, centrado, con un borde suave que le da énfasis sin romper el estilo limpio.

Debajo del texto, hay un botón prominente con el mensaje “Enviar correo”. Este botón tiene un diseño redondeado, ocupa un 25% del ancho del cuadro principal y está claramente separado del texto superior para facilitar la acción. La jerarquía visual está

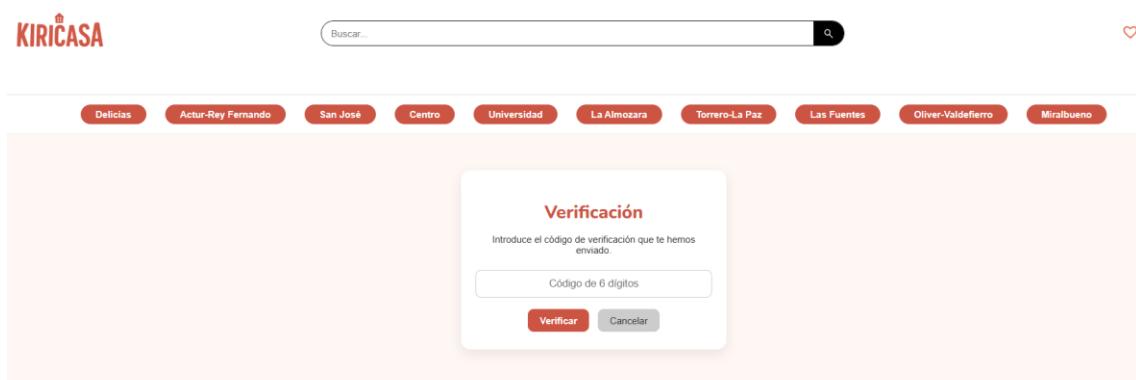


Ilustración 52 : vista solicitud de verificacion correo

bien definida, ayudando al usuario a enfocarse en el paso siguiente sin distracciones. Todo está diseñado con un enfoque responsivo y accesible.

Blog

El blog se basará en 3 filas, cada una con 10% de margen en todas las direcciones. Después, cada elemento tendrá 10% de separación lateral y un 20% de margen superior e inferior. El diseño busca optimizar el espacio visual, creando un equilibrio armónico entre los elementos que componen el blog. Esta estructura no solo facilita la lectura, sino que tambien mejora la estética general, permitiendo que cada sección respire y se destaque por sí misma. Al implementar un diseño tan meticuloso, los lectores disfrutarán de una experiencia más agradable, donde el contenido se presenta de manera clara y accesible, incentivando la interacción y el regreso al blog para futuras visitas.

Blog de Kiricasa

Consejos para encontrar piso en Zaragoza

Publicado el 15 de mayo de 2025

Buscar piso en Zaragoza puede ser todo un reto si no sabes por dónde empezar. La ciudad cuenta con una gran variedad de barrios, cada uno con sus propias ventajas. En Kiricasa recomendamos comenzar por zonas como **Actur** (ideal para familias y estudiantes del Campus Río Ebro), **Universidad** (ambiente joven y buena conexión de transporte) o **La Almozara** (zona tranquila con buenos precios).

Establece un presupuesto mensual que incluya alquiler, suministros, transporte y gastos personales. Utiliza filtros por barrio, precio y características clave como si acepta mascotas o si está amueblado. No dudes en contactar al propietario para resolver dudas, y si es posible, agenda una visita presencial antes de tomar una decisión.

¿Qué tener en cuenta al compartir piso?

Publicado el 7 de mayo de 2025

Compartir piso no es solo una forma de ahorrar dinero, tambien puede convertirse en una experiencia positiva si se gestiona bien. Lo más importante es encontrar compañeros de piso compatibles en términos de horarios, limpieza, nivel de ruido y hábitos personales.

Recomendamos tener una charla clara desde el principio sobre las normas básicas: turnos de limpieza, gastos comunes, uso de zonas compartidas, etc. Tambien es buena idea usar aplicaciones para dividir los pagos y evitar malentendidos. En Kiricasa puedes encontrar anuncios que especifican si buscan compañeros, lo cual facilita mucho este tipo de búsqueda.

Top 5 barrios más económicos en 2025

Ilustración 53 vista blog

Términos y Condiciones

Sera aproximadamente entre el 80% y 85% del ancho total, con márgenes laterales del 10% al 15% a cada lado. En cuanto a la altura, el título "Términos y Condiciones" representa entre el 10% y el 15% del espacio disponible, seguido por el bloque de texto principal, que abarca alrededor del 60% al 70% de la altura restante, dejando márgenes superior e inferior de aproximadamente un 10% a 15% cada uno.

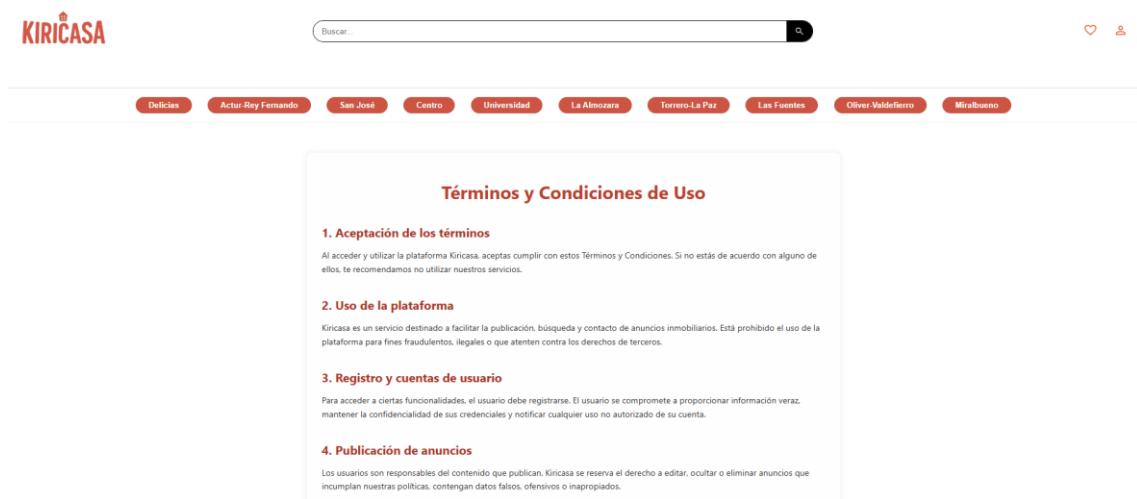


Ilustración 54 vista términos

Pestaña Enviar Mensaje

El contenido útil ocupa aproximadamente el 75% del ancho total con márgenes laterales del 12.5% a cada lado, creando un espacio limpio y organizado. Verticalmente, se divide en tres secciones principales: el título "Asunto del mensaje" ocupa entre el 15% y 20% de la altura, seguido por el bloque "Texto del usuario", que con un 50% del espacio destaca como el área principal de interacción, y finalmente los botones inferiores "Enviar" y "Enviar mensaje", que ocupan entre el 20% y 25% restante, asegurando accesibilidad y claridad en la llamada a la acción. Los márgenes internos, de un 5% a 8%



Ilustración 55 vista ver publicación

Diseño De Botones

Los botones de la página serán muy similares, todos seguirán este diseño y algunos tendrán un hover y otros no. Los que son propios de la página sí que lo tendrán, pero los que son de la cabecera no lo tendrán, como se puede ver en la imagen. Seguirán ese diseño; todos los tamaños cambiarán según el sitio donde estén. Los colores elegidos para los botones garantizarán una buena visibilidad y contraste, mejorando la accesibilidad y la estética general del sitio. Además, se incorporarán animaciones sutiles para los botones que cuenten con el efecto hover, ofreciendo una respuesta visual atractiva al interactuar con ellos. En la cabecera, el enfoque será minimalista, priorizando la funcionalidad y la navegación rápida. Así, el diseño no solo será visualmente atractivo, sino también intuitivo y eficiente para todos los usuarios.

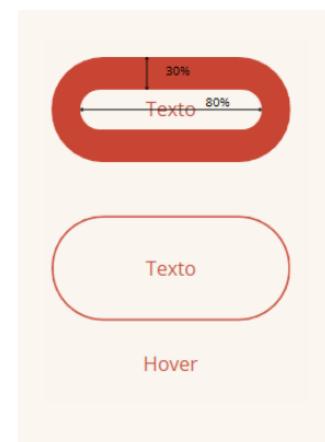


Ilustración 56 diseño de botones

Pruebas

En esta sección vamos a realizar varias pruebas con respecto al inicio de sesión, registro, publicar un anuncio, modificar perfil y modificar un anuncio...

Inicio de Sesión:

Probamos con un usuario y/o contraseña aleatoria

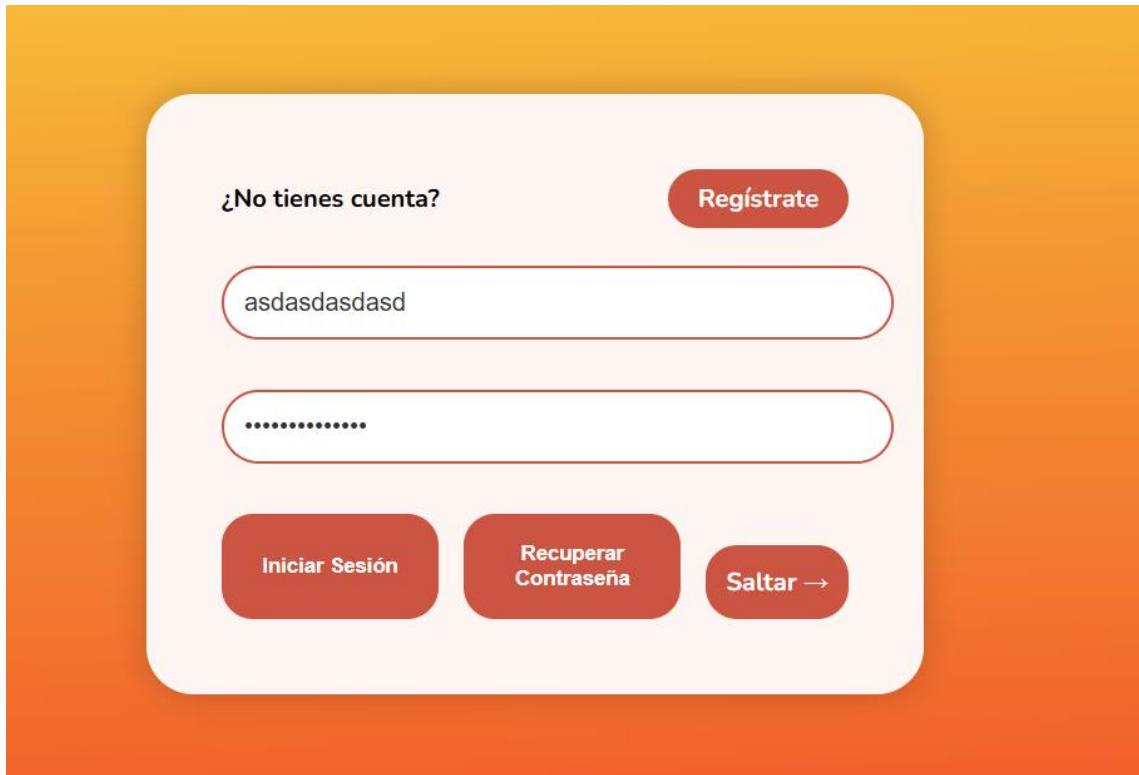


Ilustración 57 vista login

Al hacer esto observamos un mensaje de error

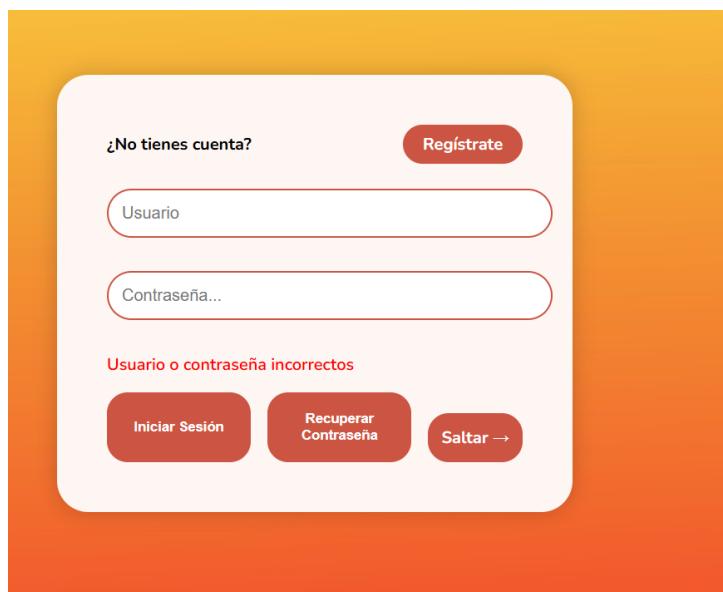
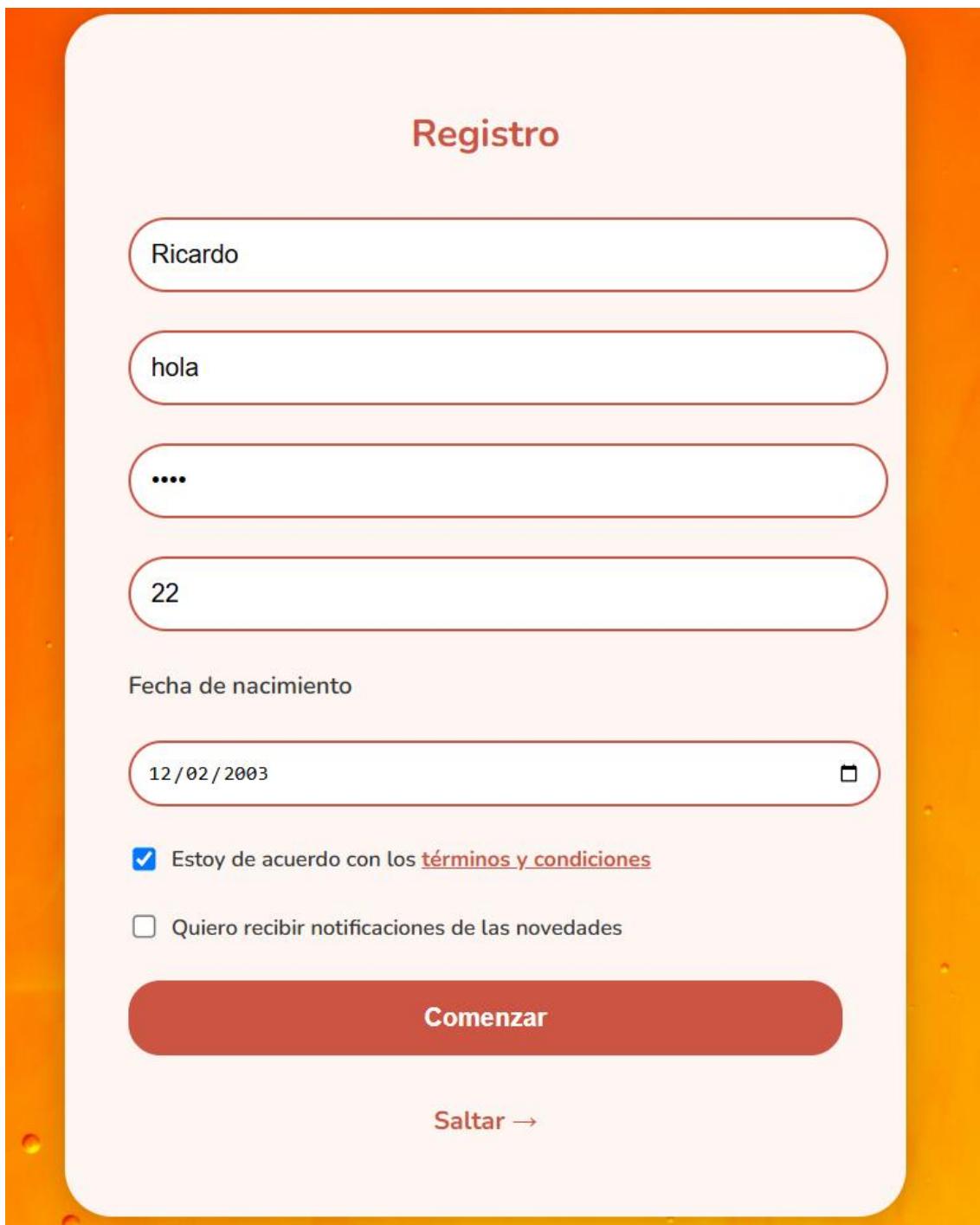


Ilustración 58 vista login

Registro

En esta prueba probaremos a no introducir un correo ni una contraseña válidos, hay que tener en cuenta que todos los campos son obligatorios.



The image shows a registration form titled "Registro". It consists of several input fields and checkboxes. The first four fields are filled with placeholder text: "Ricardo" in the first name field, "hola" in the last name field, "...." in the middle name field, and "22" in the age field. Below these is a "Fecha de nacimiento" (Date of birth) field containing "12/02/2003" with a calendar icon. Two checkboxes are present: one checked ("Estoy de acuerdo con los [términos y condiciones](#)") and one unchecked ("Quiero recibir notificaciones de las novedades"). A large red "Comenzar" button is at the bottom, and a "Saltar →" link is below it. The entire form is set against a white background with rounded corners, framed by an orange border.

Registro

Ricardo

hola

....

22

Fecha de nacimiento

12/02/2003

Estoy de acuerdo con los [términos y condiciones](#)

Quiero recibir notificaciones de las novedades

Comenzar

[Saltar →](#)

Ilustración 59 vista registro

Observamos que se queja del correo:

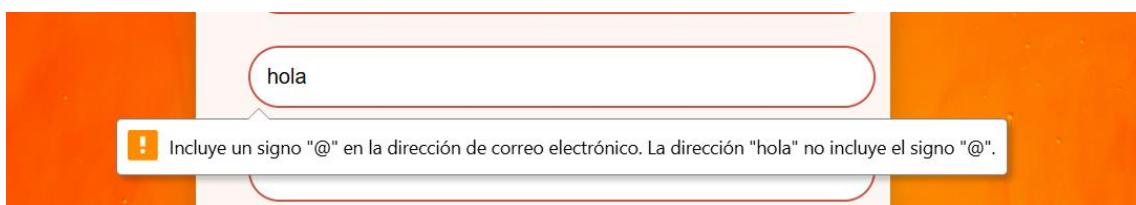


Ilustración 60 vista campo registro

El siguiente campo que da error es la contraseña ya que no cumple con la longitud solicitada.

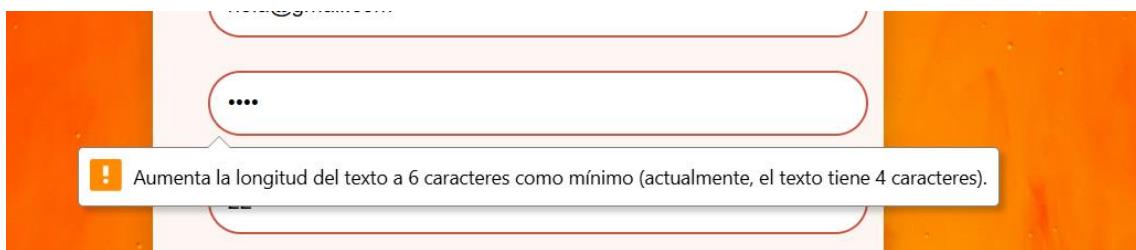


Ilustración 61 vista campo registro

El siguiente campo sería el teléfono ya que no cumple con el formato que yo solicito.

The illustration shows a registration form titled "Registro". The form includes fields for Nombre, Email, Contraseña (mín. 6 caracteres), Número de teléfono, Fecha de nacimiento (with a date input field dd/mm/aaaa), and two checkboxes for terms and conditions and notifications. At the bottom are "Comenzar" and "Saltar →" buttons. A red validation message box is displayed above the phone number field with the text: "El número de teléfono debe contener entre 9 y 15 dígitos."

Ilustración 62 vista registro

Siguiendo con las comprobaciones la siguiente es la fecha:

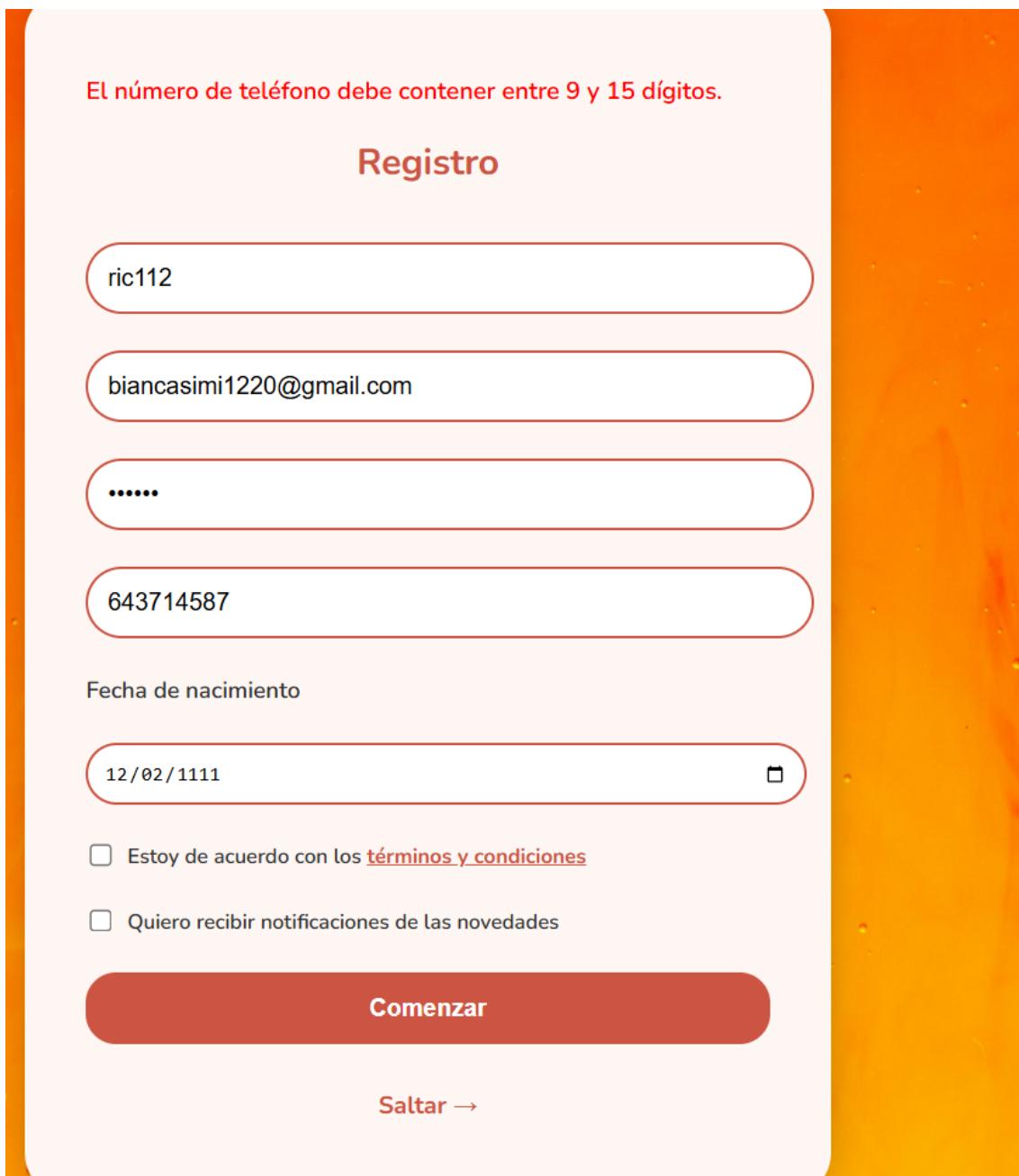


Ilustración 63 vista registro

Con este mensaje de error:

La fecha de nacimiento no puede ser anterior a 1900

Registro

Ilustración 64 vista registro error

The screenshot shows a registration form with a red header bar containing the word "Registro". Below it is a light gray input field labeled "Fecha de nacimiento" with a placeholder "dd/mm/aaaa". Inside the field, the value "12/12/2220" is displayed. To the right of the field is a small square icon with a minus sign. Below the field, there is a validation message: "El valor debe ser igual o anterior a 27/05/2025." This message is enclosed in a white box with a thin gray border and has an exclamation mark icon inside. There are two checkboxes below the message: one checked with the label "Estoy de acuerdo con los términos y condiciones" and another unchecked with the label "Quiero recibir notificaciones de las novedades". At the bottom is a large orange button labeled "Comenzar".

Ilustración 65 vista registro error

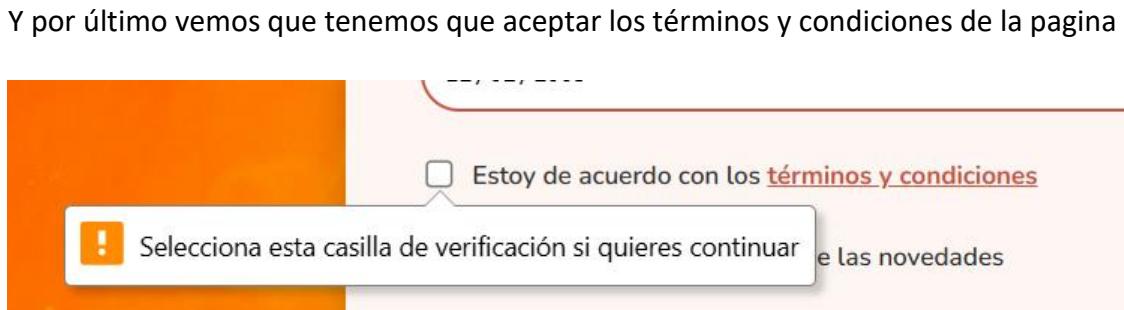


Ilustración 66 vista registro error

La siguiente prueba es intentar insertar un usuario ya existente, para lo que:

	email	nombre
▶	luis.mares@hotmail.com	lourdes.nazario
	ignacio.galvan@hotmail.com	laura.menchaca
	juancarlos.concepcion@gmail.com	juana.sisneros
	sergio.altamirano@hotmail.com	antonio.pulido
	lorena.hernandez@hotmail.com	cristina.vázquez
	rafael.caban@hotmail.com	rubén.mojica
	antonia.villasenor@hotmail.com	maríaelena.reyes
	berta.pacheco@gmail.com	amalia.cotto
	isabela.rolon@gmail.com	graciela.rosado
	recuricardo@gmail.com	ric11
	admin@kircasa.com	admin

Ilustración 67 salida bbdd usuarios

Introduciremos estos datos:

The screenshot shows a registration form titled "Registro". It contains four input fields with placeholder text: "ric11", "recurso@gmail.com", ".....", and "643714512". Below these is a date input field showing "12/02/2003". There are two checkboxes: one checked ("Estoy de acuerdo con los términos y condiciones") and one unchecked ("Quiero recibir notificaciones de las novedades"). At the bottom are two buttons: a large red "Comenzar" button and a smaller "Saltar →" button.

Ilustración 68 vista registro

Y obtendremos esta serie de errores dependiendo de que campo sea el afectado.

The illustration shows a registration form with a red error message at the top: "El email ya está registrado". The form fields are: Nombre, Email, Contraseña (mín. 6 caracteres), Número de teléfono, Fecha de nacimiento, and dd/mm/aaaa. At the bottom is a checkbox for accepting terms and conditions.

Ilustración 70 vista registro error

The illustration shows a registration form with a red error message at the top: "El nombre de usuario ya está en uso". The form fields are: Nombre, Email, and Contraseña (mín. 6 caracteres).

Ilustración 69 vista registro error

Búsqueda

Para este apartado solo buscara por direcciones y nombres de la publicación:

Por lo que si nosotros ponemos:

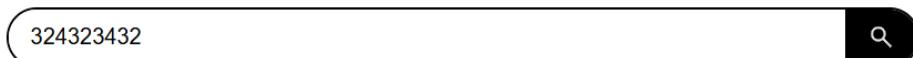


Ilustración 71 vista búsqueda

Obtendremos esto:

Resultados para: 324323432

Precio máx.

Permite mascotas

Todos los barrios
Aplicar filtros

No se encontraron resultados.

Ilustración 72 vista búsqueda

Y lo mismo para el resto de campos si se ponen caracteres inválidos o letras en el campo de habitaciones...

KIRICASA

♥

Delicias
Actur-Rey Fernando
San José
Centro
Universidad
La Almozara
Torre-La Paz
Las Fuentes

Oliver-Valdefierro
Mirabueno

Resultados para: null

Precio máx.

Permite mascotas

Todos los barrios
Aplicar filtros

No se encontraron resultados.

Ilustración 73 vista búsqueda

Publicar Anuncios

Los campos en los que se espera números no se le permite al usuario introducir letras

Precio:

Ilustración 74 vista publicar anuncio

Esto es así para precios, metros cuadrados, habitaciones y numero de compañeros...

Además, que los campos como Barrios y tipo de piso serán limitados ya que solo se podrá seleccionar una de entre las opciones yo permito:

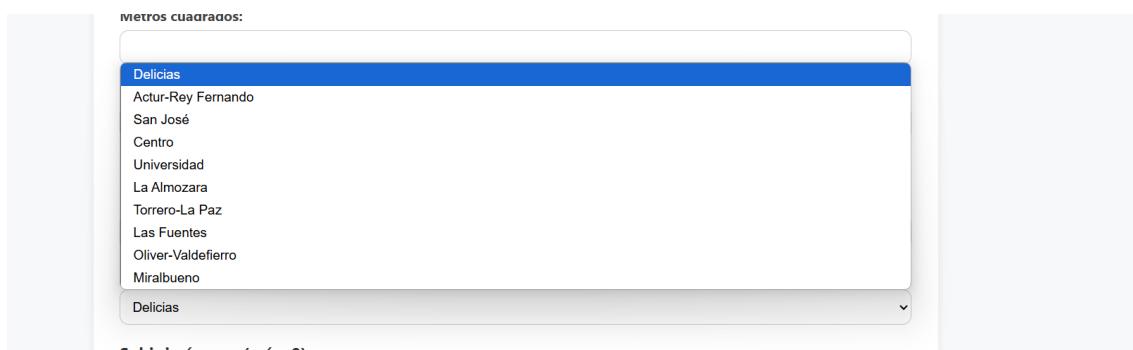


Ilustración 75 vista publicar anuncio

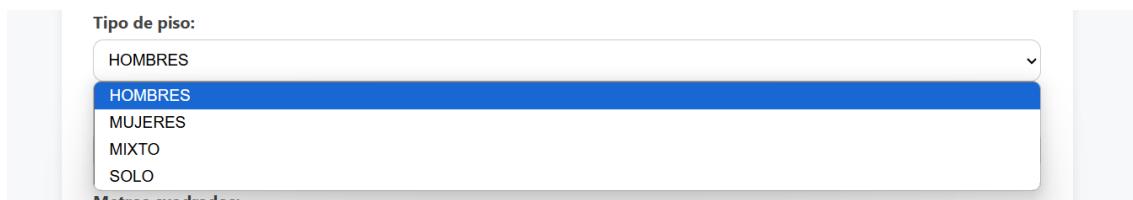


Ilustración 76 vista publicar anuncio

Además, que solo se podrán publicar archivos con extensión de fotos.

Modificar datos usuarios

Por ejemplo, si intentamos quitar el correo:

The screenshot shows a user modification form with the following fields and errors:

- Nombre:** ric11
- Email:** recuricardogmail.com
- Fecha de nacimiento:** 6437145
- Notificaciones:** Deseo recibir notificaciones por email

A validation message is displayed below the email field: **! Incluye un signo "@" en la dirección de correo electrónico. La dirección "recuricardogmail.com" no incluye el signo "@".**

At the bottom are two buttons: **Guardar cambios** (in red) and **Cancelar**.

Ilustración 77 vista modificar usuario

The screenshot shows a user modification form with the following fields and errors:

- Número de teléfono:** 64371452323123
- Fecha de nacimiento:** dd/mm/aaaa
- Notificaciones:** Deseo recibir notificaciones por email

A validation message is displayed next to the date of birth field: **! Utiliza un formato que coincida con el solicitado**.

At the bottom are two buttons: **Guardar cambios** (in red) and **Cancelar**.

Ilustración 78 vista modificar usuario

Para la edad:

Fecha de nacimiento:

!

Deseo recibir ! El valor debe ser igual o posterior a 01/01/1900.

Guardar cambios
Cancelar

Ilustración 79 vista modificar usuario

Modificar Publicación

Al igual que el formulario de publicación este tiene los mismos campos protegidos, al igual que las imágenes, solo se permite archivos con cierta extensión...

```
<input type="file"
       name="imagen"
       ...
       accept="image/png, image/jpeg, image/jpg"
       style="display: none;" 
       onchange="this.closest('form').submit()" />
</label>
```

Ilustración 80 vista admisión archivos

Por lo que sigue la misma lógica que el apartado que anteriormente se ha mencionado.

Recuperar Contraseña

Al colocar un correo aleatorio en el campo hará una comprobación y saldrá el siguiente mensaje de error



Ilustración 81 vista recuperar contraseña

Manual de Usuario

¿Qué es Kiricasa?

Es una página web diseñada para publicar y poder ponerse en contacto con gente que publica sus pisos, está pensando para el alquiler de habitaciones, pero también ofrece la oportunidad de poner pisos a la venta con un sistema de mensajes mediante correo electrónico.

Registro y acceso

1. Accede a la página principal.

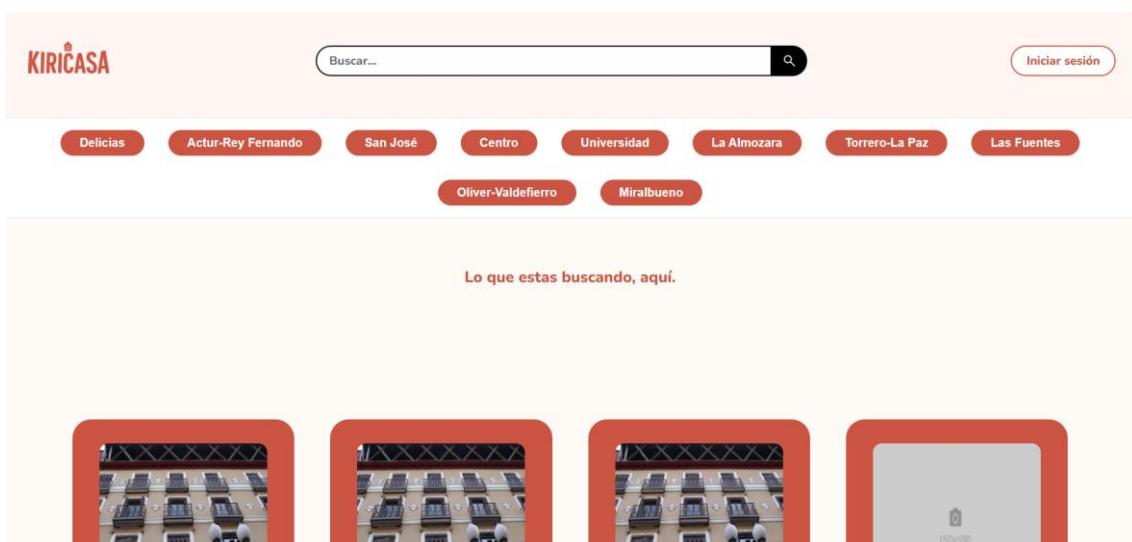


Ilustración 82 vista home no logged

2. Haz clic en "Iniciar Sesión".

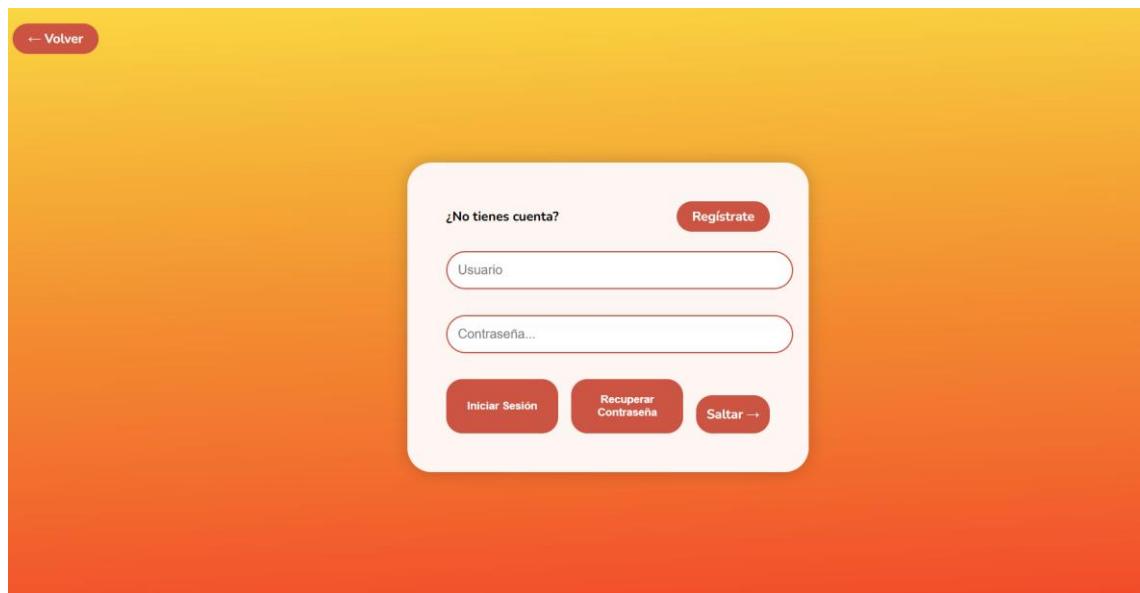


Ilustración 83 vista login

3. Haz clic en "Registrarse".

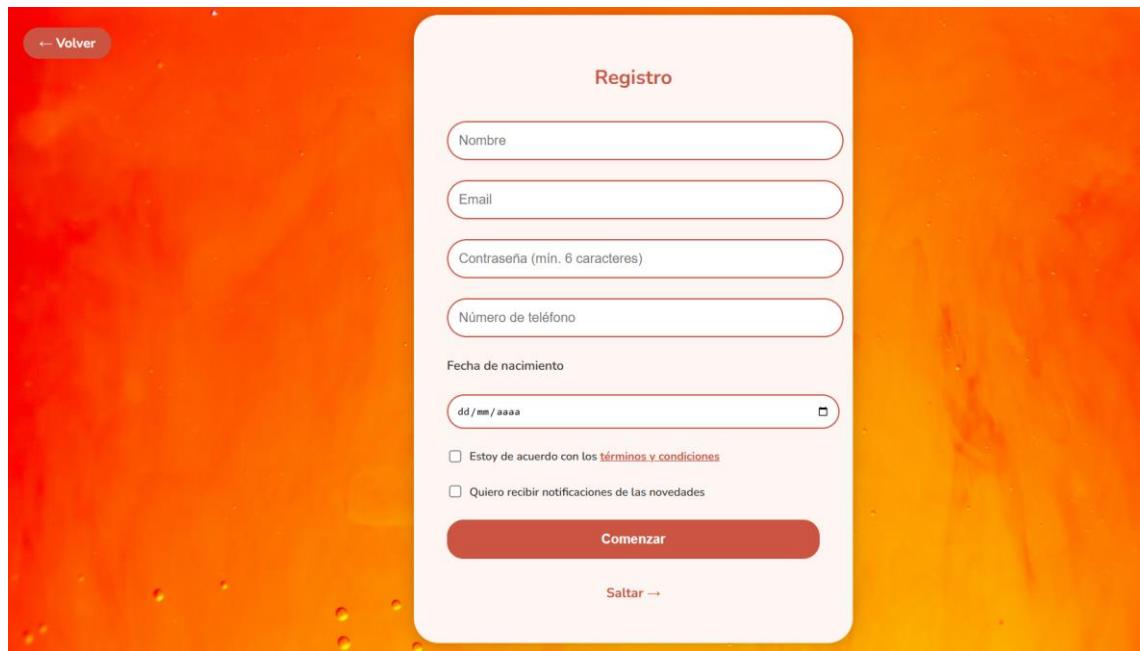


Ilustración 84 vista registro

4. Rellena el formulario con tu nombre, correo electrónico y contraseña...
5. Para mayor seguridad verifica tu cuenta...
6. Accede a la vista del perfil



Ilustración 85 vista panel de usuario

7. Clica en “Verificar correo”

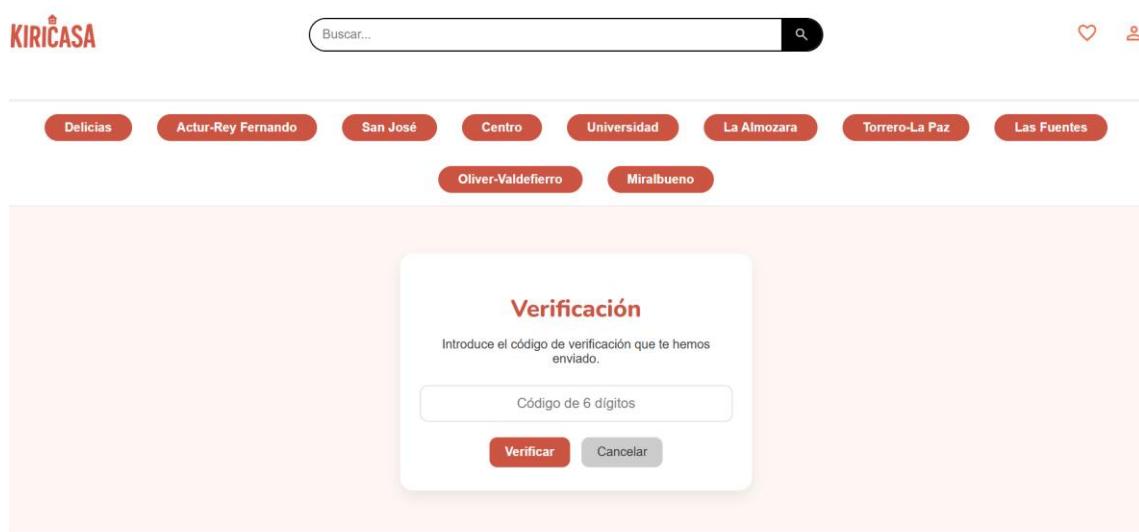


Ilustración 86 vista verificar correo

Código de verificación - Kiricasa Recibidos x



recuricardo@gmail.com
para mí ▾

10:51 (hace 0 minutos)



Tu código de verificación es: 235907

Ilustración 87 vista correo

8. Introduce el código

9. Listo ya puede disfrutar de la página con seguridad

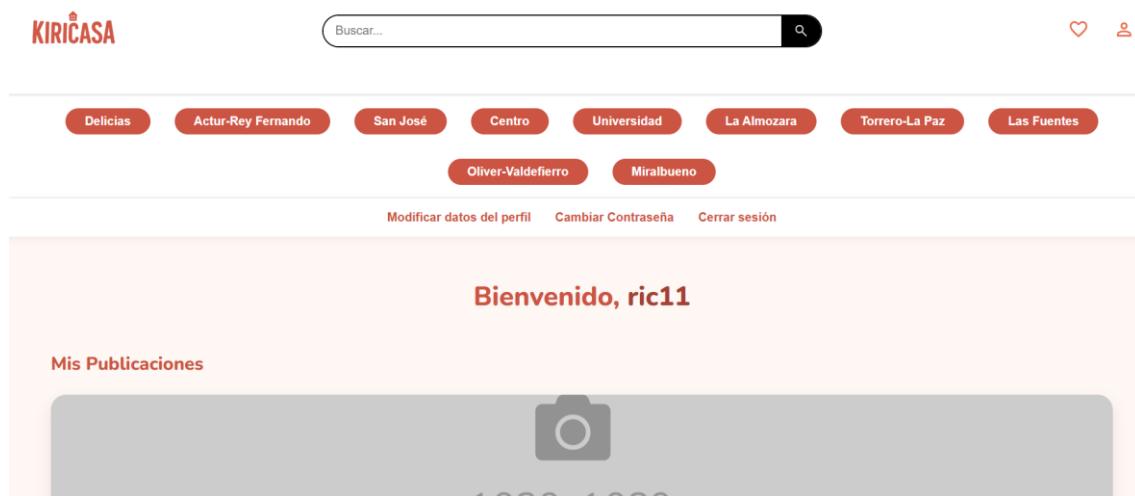


Ilustración 88 vista panel de usuario

Caso: He olvidado Mi contraseña/Modificar Contraseña

Cambiar Contraseña

En la vista de perfil tenemos una opción de modificar mi contraseña, allí se puede realizar esta acción.



Ilustración 89 vista cambiar contraseña

He olvidado mi contraseña

En el login tenemos un botón “Recuperar Contraseña”



Ilustración 90 vista recuperar contraseña

Que nos enviara un código y poniéndolo.



¡Y luego a iniciar sesión!

Caso: Publicar Anuncio

1. En la pagina principal clicamos en “Publicar Anuncio”

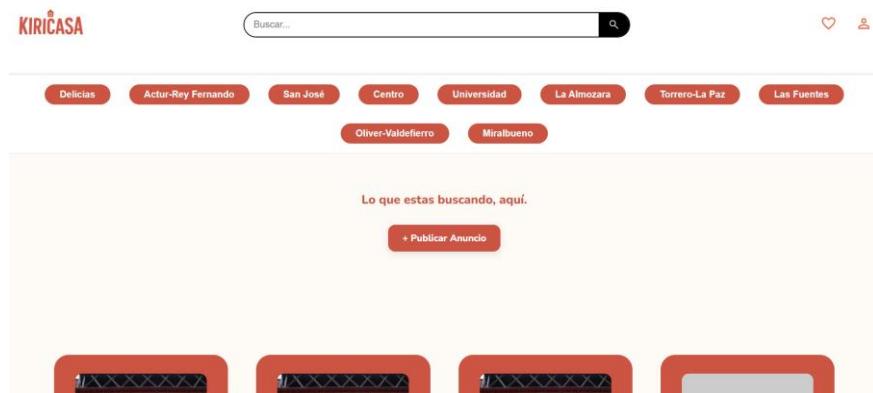


Ilustración 91 vista home

2. Rellenamos los datos

Publicar nuevo anuncio

Título:

Descripción:

Precio:

Estado:

Tipo de piso: HOMBRES

Ubicación:

Metros cuadrados:

Ilustración 92 vista publicar anuncio

Una vez finalizado y las fotos colocadas se le da a Publicar Anuncio.

Barrio:

Delicias

Subir imágenes (máx. 9):

Imagen 1 Imagen 2 Imagen 3

[Seleccionar archivo] Capture001.png [Seleccionar archivo] Ningún...cionado [Seleccionar archivo] Ningún...cionado

Imagen 4 Imagen 5 Imagen 6

[Seleccionar archivo] Ningún...cionado [Seleccionar archivo] Ningún...cionado [Seleccionar archivo] Ningún...cionado

Imagen 7 Imagen 8 Imagen 9

[Seleccionar archivo] Ningún...cionado [Seleccionar archivo] Ningún...cionado [Seleccionar archivo] Ningún...cionado

Publicar anuncio Cancelar

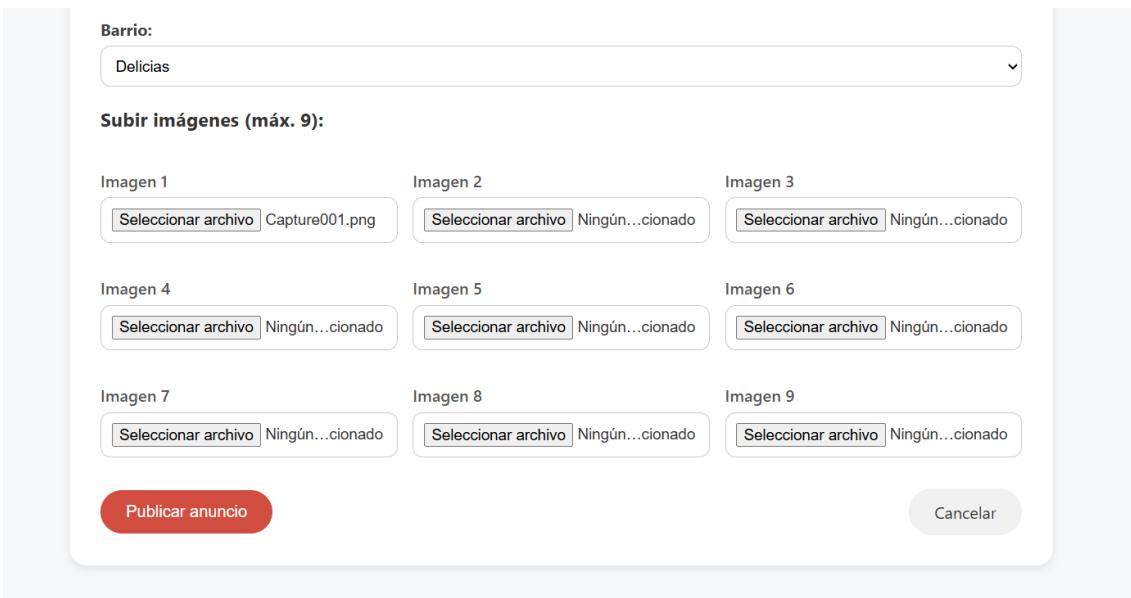


Ilustración 93 vista publicar anuncio

Caso: Modificar Anuncio

Una vez tengamos publicado el anuncio accedemos a nuestro perfil y vamos a Mis Publicaciones



Ilustración 94 vista ver perfil

Seleccionamos el anuncio que se quiere modificar.

Clicamos en “Modificar anuncio”

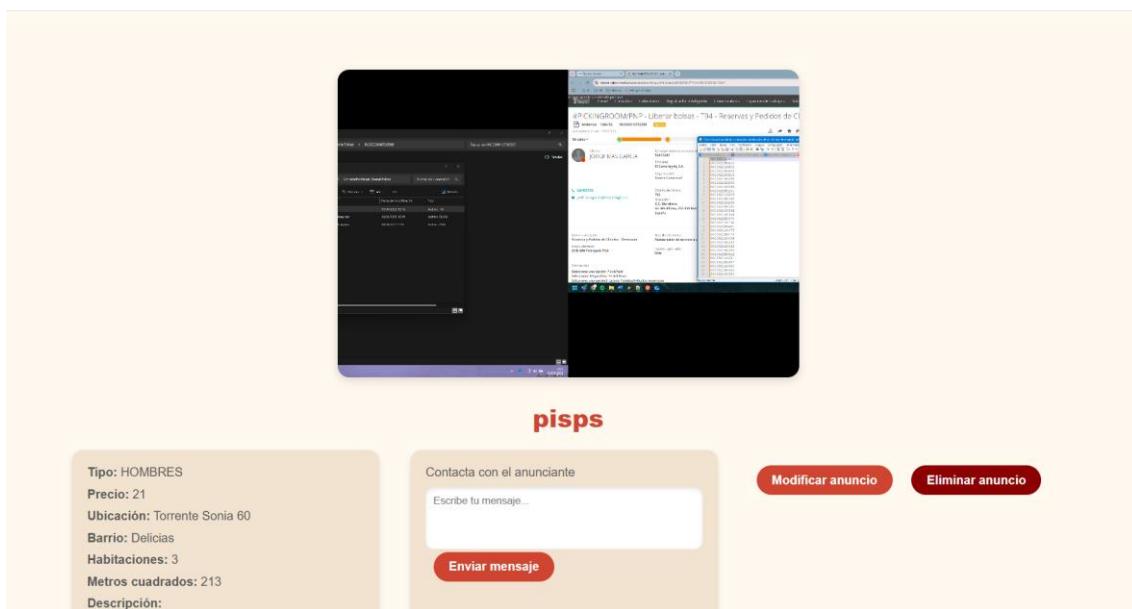


Ilustración 95 vista ver perfil

Ya qui se puede ver que tenemos varias opciones modificamos los campos que queramos o las imágenes y guardamos los campos con el botón

Número de compañeros:
1
Barrio:
Delicias
Guardar cambios

Ilustración 96 vista guardar cambios de publicación

Manual De Administrador

En la base se creará siempre un usuario con Admin y PW 1234 para que siempre haya un administrador en la página.

Como ya se ha mencionado antes los admins están pensados para que sean usuarios “normales” pero ciertos permisos, por lo que por ejemplo en la vista de una publicación ellos podrán:

Prado, Maestas y Jaimes Asociados

Tipo: HOMBRES Precio: 1049 Ubicación: Manzana Julia Yáñez, 41 Esc. 344 Barrio: Miralbueno Habitaciones: 2 Metros cuadrados: 77 Descripción: Consequatur est consequatur laboriosam dolor. Dolores exercitationem voluptas rem. Iure unde dolore est rem hic illio. Cum a labore velit blanditiis adipisci.	Contacta con el anunciante Escribe tu mensaje... <input type="button" value="Enviar mensaje"/>	<input type="button" value="Añadir a Favoritos"/>
---	--	---

Ilustración 97 vista ver publicación admin

Modificar o eliminar publicaciones.

Y en la pestaña de administración tendrán una lista con todos los usuarios:

Panel de Administración				
Gestión de Usuarios				
ID	Nombre	Email	Rol	Acciones
2	lourdes.nazario	luis.mares@hotmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
3	laura.menchaca	ignacio.galvan@hotmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
4	juana.sisneros	juancarlos.concepcion@gmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
5	antonio.pulido	sergio.altamirano@hotmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
6	cristina.vázquez	lorena.hernandez@hotmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
7	rubén.mojica	rafael.caban@hotmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
8	maríaelena.reyes	antonia.villasenor@hotmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
10	amalia.cotto	berta.pacheco@gmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>

Ilustración 98 vista panel admin

En los cuales ellos podrán hacer o quitar de administrador a los usuarios que ellos deseen, así como borrarlos...

9	marideerna.reyes	antonia.villasenor@hotmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
10	amalia.cotto	berta.pacheco@gmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>
11	graciela.rosado	isabela.rolon@gmail.com	ADMIN	<button>Borrar</button> <button>Quitar Admin</button>
12	ric11	recuricardo@gmail.com	ADMIN	<button>Borrar</button> <button>Quitar Admin</button>
14	asd	biancasimi10111@gmail.com	USER	<button>Borrar</button> <button>Hacer Admin</button>

Ilustración 99 vista panel admin

La aplicación no te dejara borrarte a ti mismo y ni quitarte de administrador.

Seguiremos con la opción de añadir un nuevo barrio

Añadir Nuevo Barrio

Nombre del barrio:

Descripción:

Ubicación:

Crear Barrio

Ilustración 100 vista panel admin

Esto está diseñado para que si se cambia la ciudad donde está la aplicación poder

Añadir Nuevo Barrio

Nombre del barrio:

Casetas

Descripción:

El barrio de los ricos de zaragoza

Ubicación:

Zaragoza pues |

Crear Barrio

Ilustración 101 vista panel admin

modificarlo de manera sencilla.



Ilustración 102 vista filtro barrios

Al añadirlo aparecerá de forma dinámica en la lista de los barrios por los que se puede filtrar.

Al igual que si lo eliminamos:

9	Oliver-Valdefierro	Zona en transformación con servicios y viviendas amplias	Calle Lagos de Coronas	Borrar
10	Miralbueno	Barrio residencial y tranquilo en expansión	Camino del Pilón	Borrar
12	Casetas	El barrio de los ricos de Zaragoza	Zaragoza pues	Borrar

Ilustración 103 vista panel admin



Ilustración 104 vista header admin

Eso sería todo el rol de Administrador.

Manual De Despliegue

Despliegue en local

1. Requisitos del sistema

Antes de iniciar el despliegue, asegúrate de tener instalado en el equipo:

- Java 17 JDK
- Maven 3.8+
- MySQL o PostgreSQL (según la base de datos usada)

- Un editor (opcional): IntelliJ IDEA, Eclipse o VS Code
- Acceso a internet (para dependencias y librerías)

2. Configuración de base de datos

1. Inicia tu gestor de base de datos (MySQL/PostgreSQL).
2. Crea una base de datos con el nombre correspondiente (por ejemplo: kiricasa).
3. Configura el usuario y contraseña para esa base de datos.
4. En el archivo application.properties (o application.yml), modifica:

```
spring.datasource.url=jdbc:mysql://localhost:3306/kiricasa
```

```
spring.datasource.username=TU_USUARIO
```

```
spring.datasource.password=TU_CONTRASEÑA
```

```
spring.jpa.hibernate.ddl-auto=update
```

3. Clonar o copiar el proyecto

1. Copia el proyecto al equipo de trabajo o clónalo desde el repositorio Git.
2. Abre el proyecto con tu editor Java.

4. Generar el JAR ejecutable

Abre una terminal en la raíz del proyecto y ejecuta:

```
mvnw.cmd clean package
```

Esto te creara un .jar → target/kiricasa-1.0.0.jar

5. Ejecutar la aplicación

Ejecuta el JAR con el siguiente comando: java -jar target/kiricasa-1.0.0.jar

La aplicación se iniciará en el puerto 8080 por defecto. Abre tu navegador y entra en:

<http://localhost:8080>

6. Acceso a la plataforma

Puedes crear un usuario desde el formulario de registro o, si tu base de datos tiene datos precargados, iniciar sesión con uno de ellos.

Despegue Online:

Opción Ngrok

1. Requisitos previos

- Git
- Ngrok (<https://ngrok.com/>)
- Java 17 y Maven correctamente instalados en tu sistema
- Tu aplicación Spring Boot funcionando localmente

2. Preparar el proyecto

Asegúrate de que tu proyecto funcione bien localmente.

Asegúrate de que el archivo **pom.xml** esté configurado correctamente y que se genere un **.jar**.

Añade estos campos en el *application.properties*

```
server.address=0.0.0.0
```

```
server.port=8080
```

3. Crear el **.jar**

`./mvnw clean package` ejecutando la aplicación después del comando

Desplegar

1. Descarga la aplicación de Ngrok
2. Autentica tu cuenta : ngrok authtoken TU_AUTHTOKEN_NGROK
3. Con tu app corriendo en el puerto 8080, lanza: ngrok http 8080

<https://abcd1234.ngrok.io> es la URL pública segura para acceder a tu Kiricasa desde cualquier dispositivo.

Opción Apache TomCat en un servidor propio o VPS

Requisitos previos

Un servidor VPS o dedicado con IP pública (por ejemplo, contratado en DigitalOcean, OVH, etc.)

Apache Tomcat 10 instalado y funcionando en el servidor

Java 17 y Maven instalados

Acceso por SSH al servidor

Conexión a MySQL remoto o instalado en el mismo servidor

El puerto 8080 (u otro que uses) abierto en el firewall

2. Preparar el proyecto

El proyecto tiene que dejar un **.war**: <packaging>war</packaging> → en pom.xml

En el **application.properties** esté bien configurada la conexión a la base de datos externa (no local), con IP pública si MySQL está en otro servidor.

3. Empaquetar la aplicación

Ejecutar ./mvnw clean package para generar el .war

4. Subir al servidor

Ruta genérica:

```
scp target/kiricasa.war usuario@IP_DEL_SERVIDOR:/ruta/a/tomcat/webapps/
```

Reemplazar **/ruta/a/tomcat/** por la ruta real de tu instalación.

5. Iniciar Tomcat en el servidor

Ejecuta este comando :

```
cd /ruta/a/tomcat/bin
```

```
./startup.sh
```

6. Acceder desde cualquier dispositivo

Una vez desplegada, cualquier persona podrá acceder desde su navegador con la IP pública o dominio del servidor: http://IP_DEL_SERVIDOR:8080/kiricasa

Conclusiones

Ha sido un reto de principio a fin. Desarrollar una aplicación completa en un lenguaje que tenía algo olvidado, como Java, y adaptar a este entorno todo lo que había aprendido previamente en PHP, ha sido una experiencia exigente. En muchos momentos el proceso ha sido más estresante que divertido, especialmente al tener que cambiar de mentalidad y acostumbrarme a una forma distinta de estructurar el código, gestionar la lógica y aplicar la arquitectura del framework Spring Boot.

Aun así, creo sinceramente que ha merecido la pena. Me ha permitido reforzar muchos conocimientos que ya tenía, aprender otros completamente nuevos y, sobre todo, mejorar mi capacidad de razonamiento lógico como desarrollador. He comprendido mejores conceptos como el enrutado, los controladores, la gestión de sesiones, la seguridad con JWT o la organización modular de un proyecto grande. Todo esto me ha hecho sentir que doy un paso adelante en cuanto a madurez técnica y autonomía.

El aprendizaje de Spring Boot ha sido más complejo de lo que esperaba. Es un framework muy potente, pero también tiene una curva de aprendizaje considerable. Esto, unido a los plazos limitados del proyecto, me ha obligado a tomar decisiones importantes, priorizar las funcionalidades clave y recortar o dejar para el futuro otras que me habría gustado implementar.

A pesar de todo ello, estoy satisfecho con el resultado. He conseguido sacar adelante una aplicación funcional, estructurada, visualmente cuidada y útil. Kiricasa no es solo un ejercicio académico, sino que podría tener una utilidad real en un contexto específico. Haberlo construido desde cero y haber llegado hasta aquí es, para mí, un logro importante y motivo de orgullo.

Bibliografía

- ChatGPT (OpenAI). Asistencia técnica y resolución de dudas sobre desarrollo web y Spring Boot. Consultado entre marzo y mayo de 2025.
- Stack Overflow. Foro de preguntas y respuestas sobre programación.
- YouTube. Tutoriales y guías en vídeo sobre Java, Spring Boot, JWT, MySQL, Thymeleaf, etc.
- Documentación oficial de Java, Spring Boot, MySQL y Thymeleaf.
- Blogs y páginas web técnicas diversas consultadas durante el desarrollo del proyecto.
- GitHub. Repositorios y ejemplos de código relacionados con el desarrollo backend y despliegue.
- Heroku. Guía básica para despliegue de aplicaciones web en la nube.
- Apache Tomcat. Guía de configuración y despliegue de aplicaciones Java.