Proyecto de Desarrollo: Revista

José Luciano Amaya Carrascal

Hector Fabio Martínez Gómez

Daniel Alejandro Rodríguez Aguirre

Julio Reinaldo Valverde Morán

Universidad Autónoma de Manizales

Manizales, Caldas

2018

Proyecto de Desarrollo: Revista

José Luciano Amaya Carrascal

Hector Fabio Martínez Gómez

Daniel Alejandro Rodríguez Aguirre

Julio Reinaldo Valverde Morán

Cristian Giovanni Castrillón Arias

Universidad Autónoma de Manizales

Manizales, Caldas

2018

**Contenido**

Pág.

Introducción 4

1. Necesidades del cliente 5

2. Alcance 6

3.Metodología 7

3.1 Las Fases de Iconix 7

3.1.1 Análisis de Requisitos 7

Requisitos Funcionales 7

Requisitos No Funcionales 9

Modelo de Dominio 10

Mockups 11

Casos de Uso 14

3.1.2 Análisis y Diseño Preliminar 16

Casos de uso detallado 16

Diagrama de Robustez 20

Diagrama de Clases 25

3.1.3 Diseño 26

Diagrama de Secuencia 26

3.1.4 Implementación 29

4. Cronograma 30

**Introducción**

Este trabajo aborda un problema en la realización de una revista electrónica para el departamento de ingeniería de sistemas de la UAM.

La solución busca maximizar la productividad del editor encargado de la revista, brindando herramientas para facilitar la gestión de los artículos, los autores y los pares evaluadores.

Además, el sistema permitirá que los lectores busquen o descarguen cualquier artículo de manera libre y gratuita.

La página contará con un diseño amigable, que le permitirá a cada usuario visualizar la información pertinente, el diseño será de fácil uso, de tipo multiusuario, es decir, que podrá ser consultado en cualquier momento por más de un usuario a la vez y desde cualquier lugar que tenga acceso a internet.

1. **Necesidades del cliente**

Este sistema de software servirá para administrar una nueva revista electrónica para el Departamento de Ingeniería de Sistemas de la UAM. La idea es que el sistema maximice la productividad del editor de la revista, brindándole un conjunto de herramientas que automaticen la evaluación de los artículos y su eventual publicación.

Los artículos serán publicados constantemente por los diferentes autores de la revista, para la correcta publicación, éstos se someterán a una rigurosa evaluación que permitirá determinar si el contenido del artículo cumple con los requisitos estipulados por la entidad a cargo de la revista (en este caso, la UAM).

En el caso de que el artículo no sea aceptado se le dará la oportunidad al autor de que realice correcciones que serán sugeridas por los pares que evaluaron el artículo.

Para la conformación del grupo de pares evaluadores, se le realizara la invitación a los autores que cumplen con los requisitos estipulados por la entidad a cargo de la revista, y que deseen unirse a este grupo.

La aplicación será de acceso público, lo que permite que cualquier usuario pueda ingresar y leer o descargar algún artículo que le interese.

1. **Alcance**

En el presente documento el grupo de desarrollo se compromete a entregar a la revista UAM una aplicación totalmente funcional, cumpliendo con todos los requisitos mencionados en el apartado 2.1.1, ya que una vez iniciada la fase de implementación no se aceptarán cambios de ningún tipo en el sistema.

1. **Metodología**

Iconix es una metodología pesada-ligera de Desarrollo del Software que se halla a medio camino entre un RUP (Rational Unified Process) y un XP (eXtreme Programming).

Iconix deriva directamente del RUP y su fundamento es el hecho de que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML, con lo cual se simplifica muchísimo el proceso sin perder documentación al dejar solo aquello que es necesario.

Esto implica un uso dinámico del UML de tal forma que siempre se pueden utilizar otros diagramas además de los ya estipulados si se cree conveniente.

Iconix se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. El objetivo es que a partir de los casos de uso se obtenga el sistema final.

* 1. **Las fases de Iconix:**

Iconix se estructura en cuatro fases. La primera de ellas es el análisis de requisitos, seguida del análisis y diseño preliminar, a continuación, viene el diseño y finaliza con su implementación.

Previamente a esto, sin embargo, deberemos realizar un pequeño storyboard de la interfaz gráfica, con dibujos de las pantallas principales del sistema a partir de las reuniones con el cliente.

* + 1. **Análisis de Requisitos**

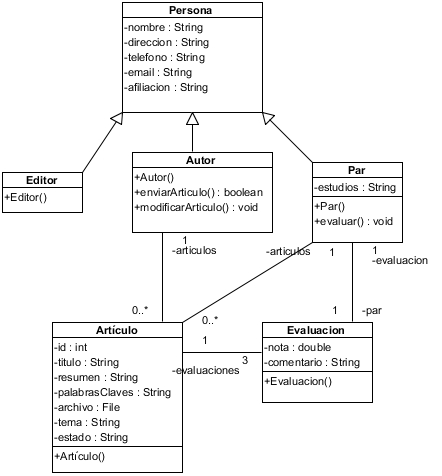
**Requerimientos Funcionales:**

* El sistema debe permitir al editor la comunicación y gestión de un grupo de pares evaluadores y autores.
* El editor puede actualizar los datos de autores, pares, evaluaciones y artículos.
* Se debe permitir al editor visualizar y acceder a los diferentes artículos.
* Se permitirá al editor asignarle 3 pares evaluadores a un artículo.
* El editor puede visualizar la evaluación de cada uno de los pares.
* Después que un artículo este revisado el editor puede enviar una notificación al autor informado el estado de su artículo además enviará los cometarios que realizaron los pares.
* El editor decide si publica el artículo en la página.
* El sistema permite a los pares evaluar los artículos asignados con una valoración de 1 a 5.
* Los pares podrán listar los artículos asignados.
* El sistema permitirá al par evaluador la retroalimentación del artículo al editor.
* Los autores podrán enviar sus artículos para ser evaluados.
* El autor puede visualizar las notificaciones enviadas por el editor.
* El sistema debe contener un formulario que el autor debe completar cuando quiera enviar un nuevo artículo (el título, temas, palabras claves, resumen del articulo y archivo a subir).
* Los lectores podrán buscar y descargar los artículos de manera gratuita.
* El sistema debe permitir el login de usuarios.
* Se debe permitir la creación de nuevas cuentas.
* Se dispondrá de un chat que permite la comunicación entre el editor y los pares.
* El sistema calculara un promedio de las calificaciones realizadas por los pares.

**Requerimientos No Funcionales:**

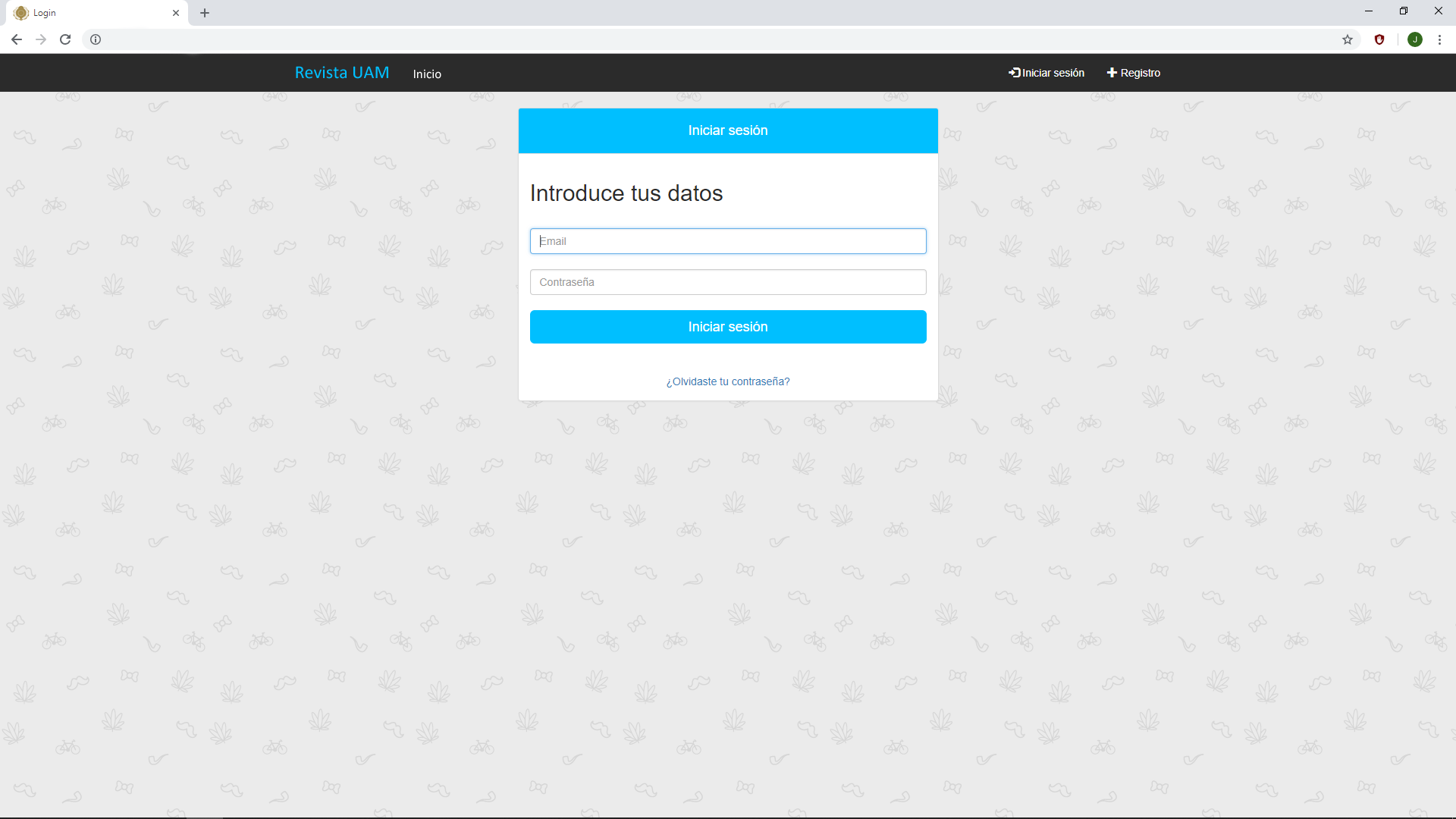
* La aplicación tratará de estar disponible el 80% de las veces que un usuario intente acceder.
* La cantidad de clientes que podrán estar conectados al tiempo será de 500.
* El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 5 horas.
* Los datos modificados en la base de datos deben ser actualizados para los usuarios en menos de 5 segundos.
* El diseño de la aplicación debe ser agradable y amigable.
* El sistema debe verificar que todos los campos de información requeridos para una acción estén diligenciados completa y correctamente.
* El sistema debe mantener informado al usuario de cada proceso que está realizando.
* Los artículos se guardarán en el disco duro del servidor en una carpeta llamada artículos.

**Modelo de Dominio**

****

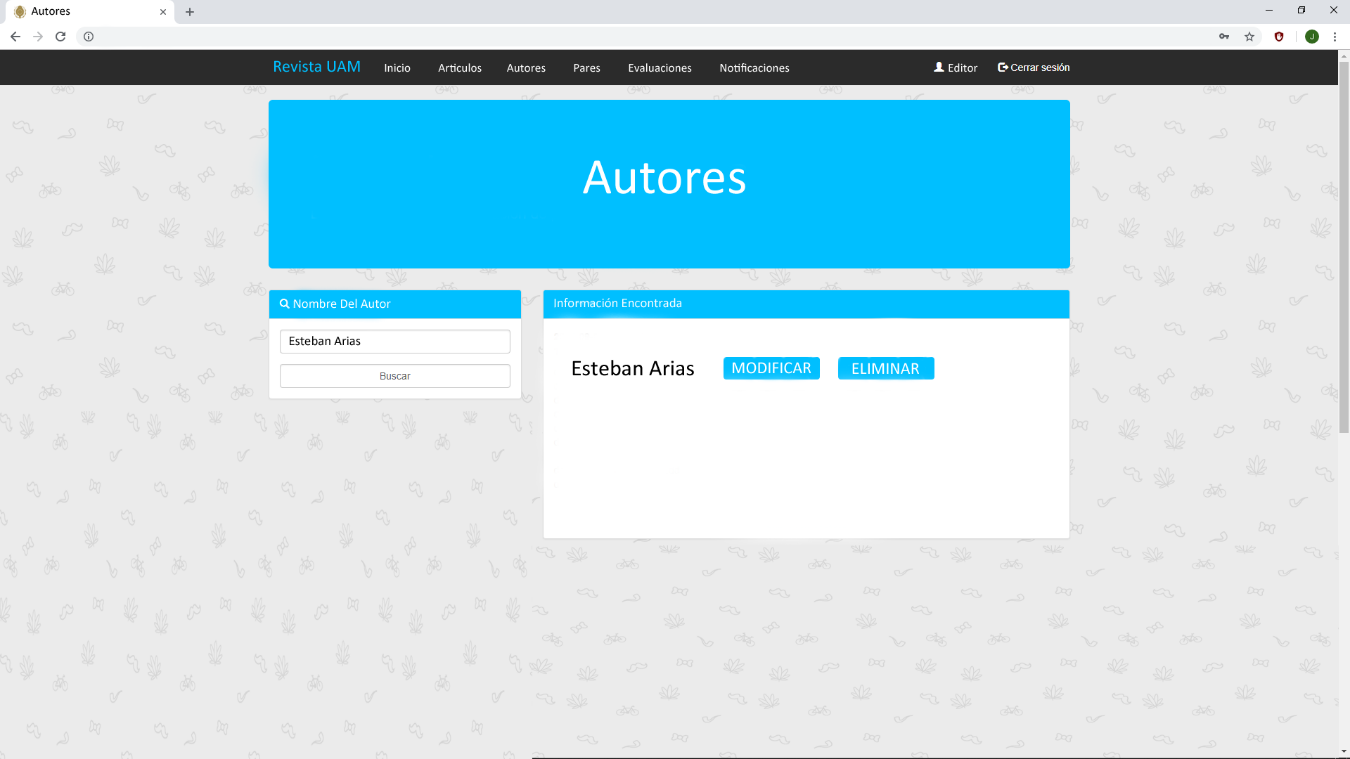
**Mockups**

**Mockup: Login de Usuario**

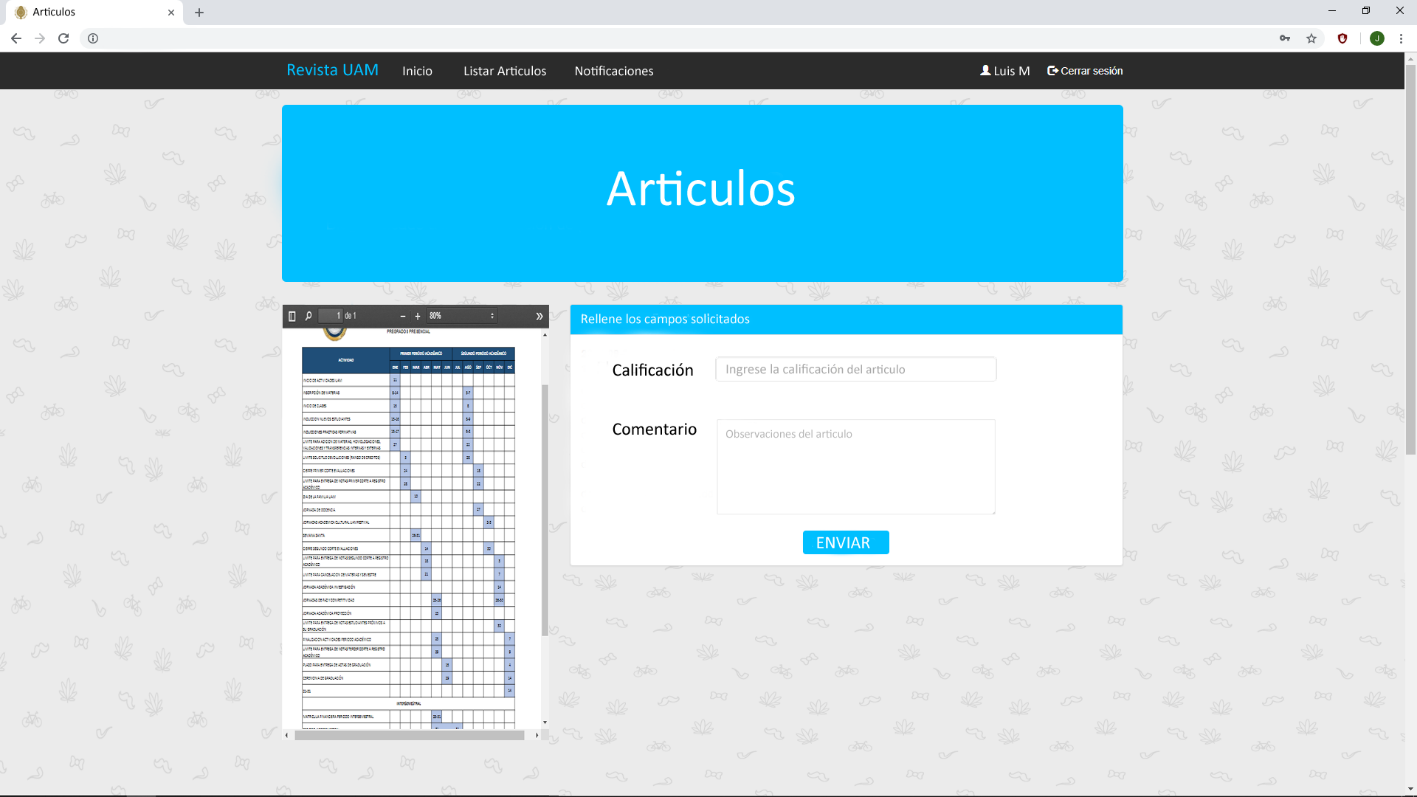
****

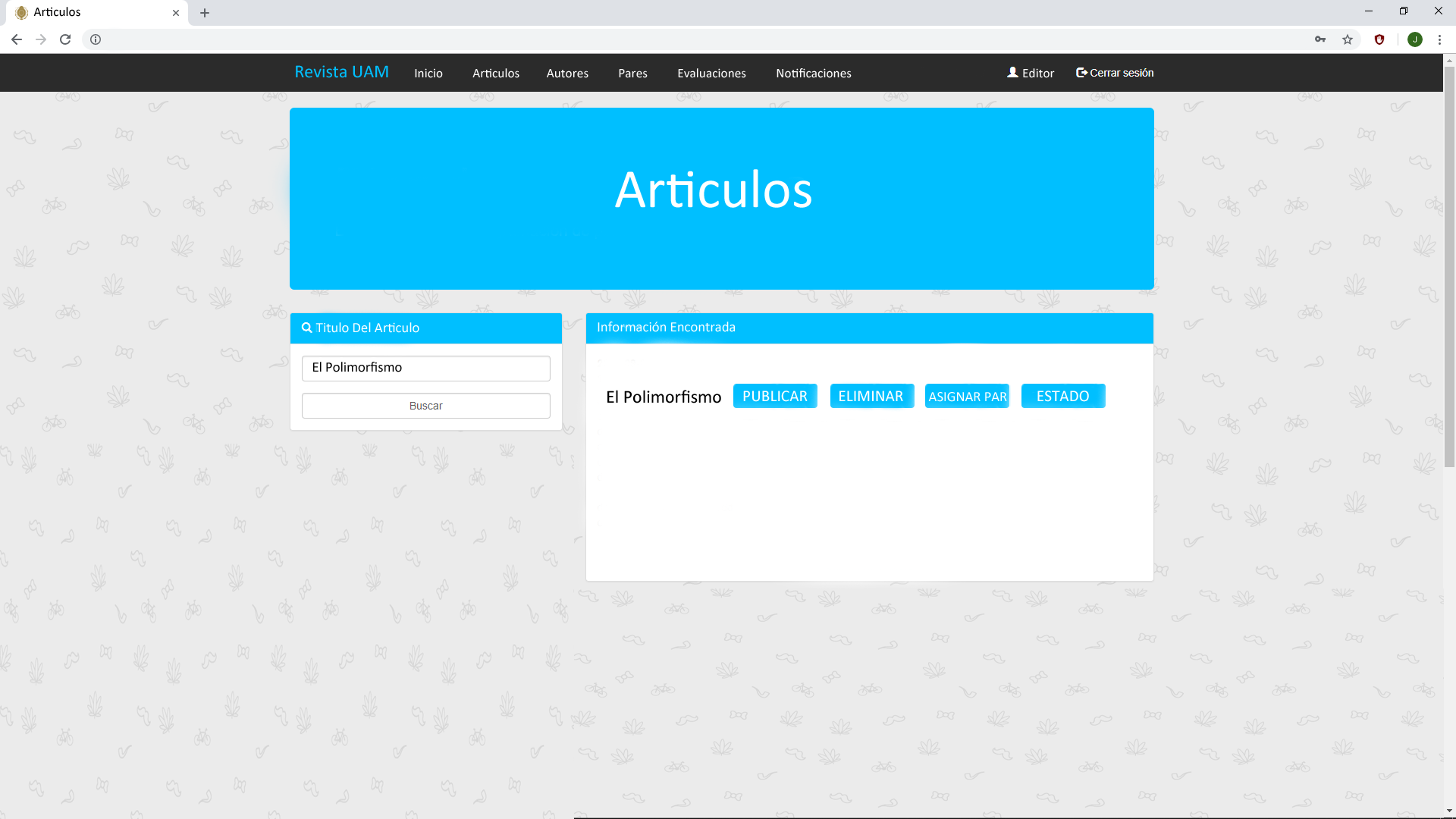
**Mockup: Registro de usuario**

**Mockup: Gestión de Autores**

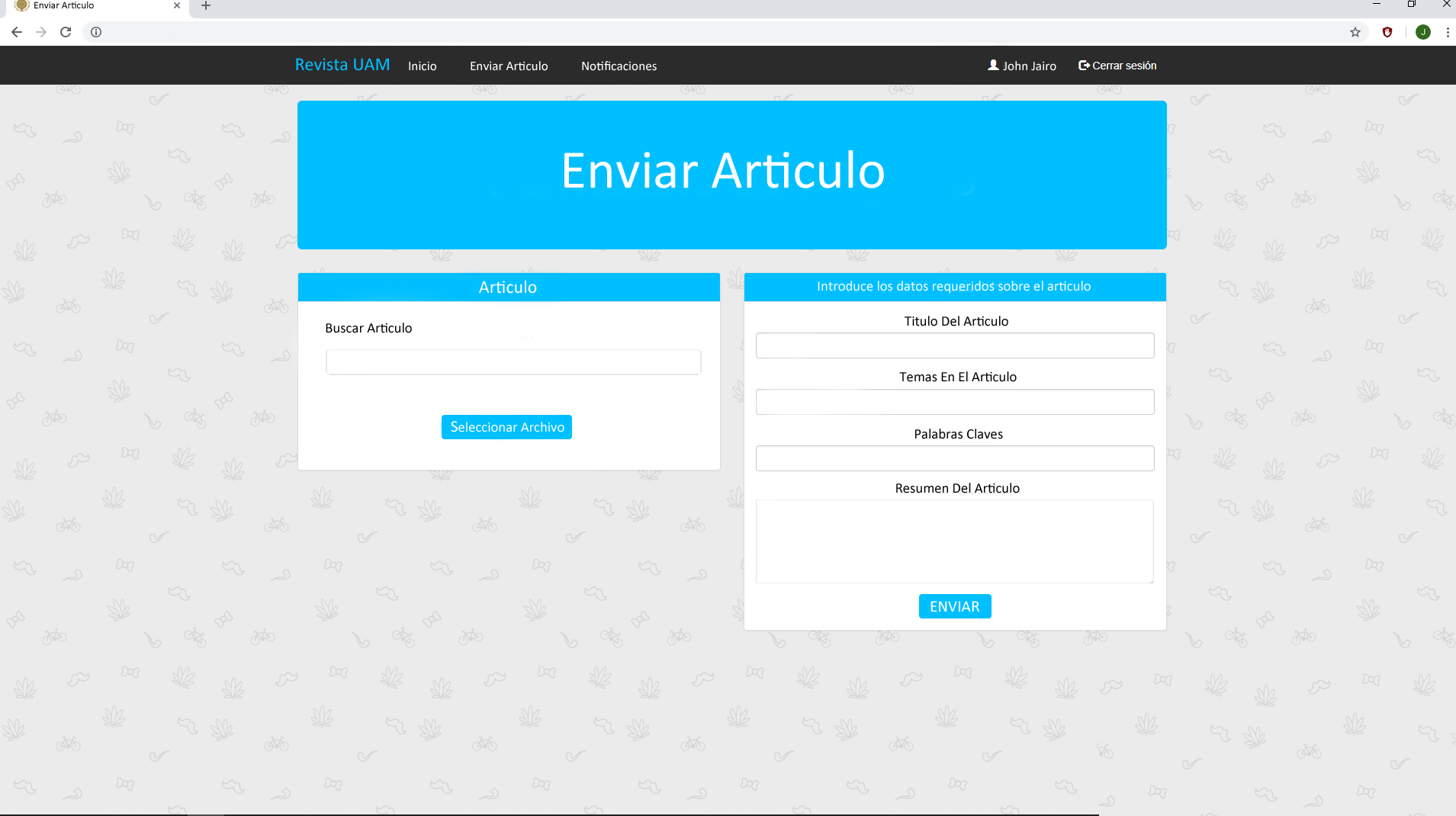
****

**Mockup: Evaluación del Articulo**

****

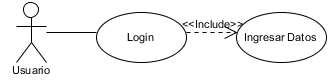
**Mockup: Gestionar Articulo**

**Mockup: Enviar Articulo**

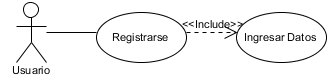
****

**Diagramas de Casos de Uso:**

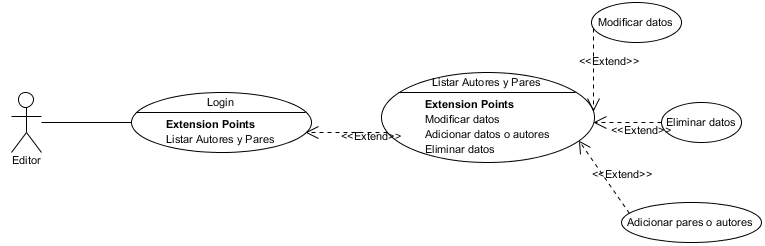
**Diagrama de casos de uso: Realizar Login**

****

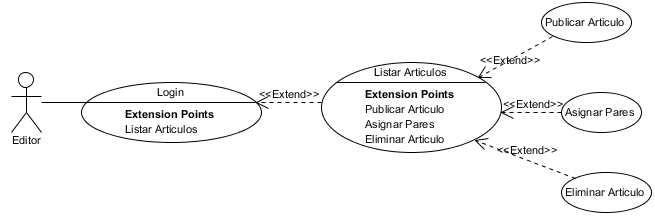
**Diagrama de casos de uso: Realizar Registro**

****

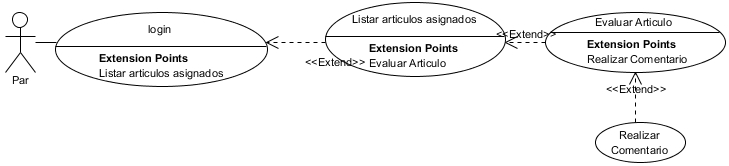
**Diagrama de casos de uso: Gestionar Autores y Pares**

****

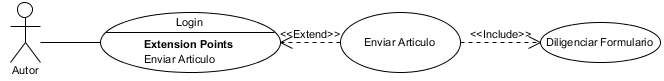
**Diagrama de casos de uso: Gestionar Artículos**

****

**Diagrama de casos de uso: Evaluar Articulo**

****

**Diagrama de casos de uso: Enviar Articulo**

****

* + 1. **Análisis y diseño Preliminar**

**Caso de uso detallado: Realizar login.**

|  |  |  |
| --- | --- | --- |
| **ID** | CU-RL-01 | |
| **Nombre** | Login | |
| **Descripción** | Permite al usuario ingresar a la página con su usuario y contraseña. | |
| **Actores** | Usuario: Editor, Par y Autor | |
| **Precondiciones** | El usuario ya se ha registrado en la página e ingresado en el apartado de login. | |
| **Poscondiciones** | Se redirecciona al usuario a la página de inicio, mostrando las diferentes opciones para cada usuario. | |
|  | | |
| **Flujo normal de eventos** | | |
| 1. El usuario selecciona en la opción de login. 2. El sistema muestra una página que contiene el formulario para loguearse. 3. El usuario diligencia el formulario de login:  * email. * contraseña.  1. El sistema redirecciona al usuario a la página de inicio. | | |
| **Flujos alternos**   * No es posible iniciar sesión con los datos ingresados por el usuario, por lo tanto, se le informa al usuario que corrija estos o si no se ha registrado, que lo haga. * El usuario no diligencio uno o los dos campos requeridos para el login, entonces el sistema muestra un mensaje informativo diciendo que se deben diligenciar todos los campos. | | |
|  | | |
| **Otros requerimientos** | |  |

**Caso de uso detallado: Realizar registro.**

|  |  |  |
| --- | --- | --- |
| **ID** | CU-RR-02 | |
| **Nombre** | Registro de usuario | |
| **Descripción** | Permite al usuario registrarse a la página. | |
| **Actores** | Usuario: Autor, Par. | |
| **Precondiciones** | El usuario ha ingresado a la página, en el apartado registro. | |
| **Poscondiciones** | Se muestra un mensaje que informa que el registro ha sido correcto y al mismo tiempo se le da la opción de ir a la página de login. | |
|  | | |
| **Flujo normal de eventos** | | |
| * 1. El usuario diligencia el formulario de registro * Nombre * Dirección * Teléfono * E-mail * Afiliación (Departamento, Institución) * Contraseña   1. El usuario da click en el botón enviar datos.   2. El sistema verifica que la información este completa.   3. El sistema guarda la información y muestra un mensaje de registro correcto. | | |
| **Flujos alternos**   * El usuario no diligencia uno o más campos del formulario y el sistema muestra un mensaje informativo y no lo deja continuar con el flujo normal hasta que todos los campos estén diligenciados. * Ya existe un usuario con el e-mail ingresado y el sistema muestra un mensaje informativo y no lo deja continuar con el flujo normal hasta que cambie el e-mail. | | |
|  | | |
| **Otros requerimientos** | |  |

**Caso de uso detallado: Gestionar autores y pares**

|  |  |  |
| --- | --- | --- |
| **ID** | CU-GAP-03 | |
| **Nombre** | Gestionar Autores y Pares | |
| **Descripción** | Permite al editor la gestión de los autores y los pares | |
| **Actores** | Usuario: Editor | |
| **Precondiciones** | El usuario ha iniciado sesión en la página. | |
| **Poscondiciones** | La actualización ha sido correcta | |
|  | | |
| **Flujo normal de eventos** | | |
| * 1. El usuario Inicia sesión y va al apartado gestionar autores o pares.   2. El sistema le da la opción de listar autores o listar pares.   3. Cuando el usuario lista los autores o los pares se le muestra la opción de actualizar los datos de estos.   4. Después de listar los autores o los pares el usuario tiene la opción de buscar un autor o un par en específico. | | |
|  | | |
| **Excepciones**   * El autor o el par que busco el usuario no se encuentra registrado en el sistema. | | |
| **Otros requerimientos** | |  |

**Caso de uso: Gestionar artículos**

|  |  |  |
| --- | --- | --- |
| **ID** | CU-GA-04 | |
| **Nombre** | Gestionar artículos. | |
| **Descripción** | Permite al editor la gestión de los artículos. | |
| **Actores** | Usuario: Editor. | |
| **Precondiciones** | El usuario ha iniciado sesión en la página. | |
| **Poscondiciones** | La gestión ha sido correcta | |
|  | | |
| **Flujo normal de eventos** | | |
| * 1. El usuario Inicia sesión y va al apartado gestionar artículos.   2. El sistema lista los artículos que han enviado los autores.   3. Una vez listados los artículos el usuario tiene la opción de publicar, eliminar o asignarle par a cada artículo.   4. Por cada opción que el usuario escoja se le mostrara un mensaje informativo avisándole del éxito de la acción. | | |
|  | | |
|  | | |
| **Otros requerimientos** | |  |

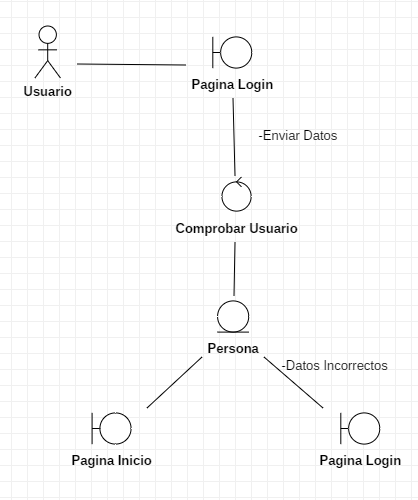
**Caso de uso detallado: Evaluar artículo**

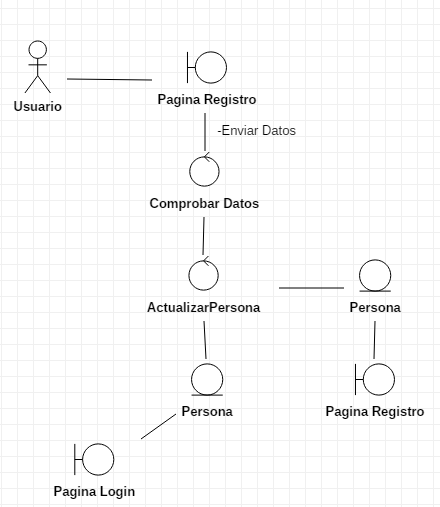
|  |  |  |
| --- | --- | --- |
| **ID** | CU-EA-05 | |
| **Nombre** | Evaluar artículos. | |
| **Descripción** | Permite a los pares realizar la evaluación de un artículo. | |
| **Actores** | Usuario: Par | |
| **Precondiciones** | El usuario ha iniciado sesión en la página. | |
| **Poscondiciones** |  | |
|  | | |
| **Flujo normal de eventos** | | |
| 1. El usuario va al apartado evaluar artículos. 2. Una vez dentro del apartado evaluar artículos, se listan los artículos que le han sido asignados para la evaluación. 3. Para cada artículo el usuario tiene la opción de dar click en el botón evaluar. 4. Cuando el usuario da click en el botón de evaluar se muestra un formulario que el usuario deberá diligenciar, el formulario contendrá:  * Calificación * Comentario (opcional)  1. La calificación que se deberá colocar deberá estar entre 1 y 5 | | |
| **Flujo alternativo**   * El usuario no diligencio el apartado de calificación, entonces se le mostrara un mensaje informándole que se debe completar esa opción. | | |
|  | | |
| **Otros requerimientos** | |  |

**Cado de uso detallado: Enviar artículo**

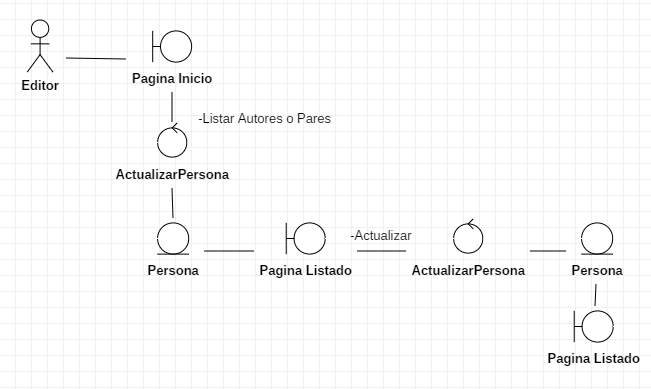
|  |  |  |
| --- | --- | --- |
| **ID** | CU-EA-06 | |
| **Nombre** | Enviar articulo | |
| **Descripción** | Permite al usuario enviar un articulo | |
| **Actores** | Usuario: Autor | |
| **Precondiciones** | El usuario ha iniciado sesión en la página web | |
| **Poscondiciones** | Se envía una notificación al editor | |
|  | | |
| **Flujo normal de eventos** | | |
| 1. El usuario selecciona la opción de enviar artículo. 2. Se le mostrará un formulario que contendrá:  * Titulo * Tema * Palabras clave (opcional) * Resumen * Seleccionar archivo  1. El sistema verifica que los campos estén correctamente diligenciados 2. Si los campos están bien, se le notificara al editor 3. El artículo se guarda en el disco duro del servidor | | |
| **Flujos alternos**   * Uno o más de los campos del formulario no se ha diligenciado y se le mostrará un mensaje informándole al usuario tal problema con la opción de corregir * La extensión del articulo seleccionado por el usuario no es válida para el sistema. | | |
|  | | |
| **Otros requerimientos** | |  |

**Diagrama de Robustez: Realizar Login**

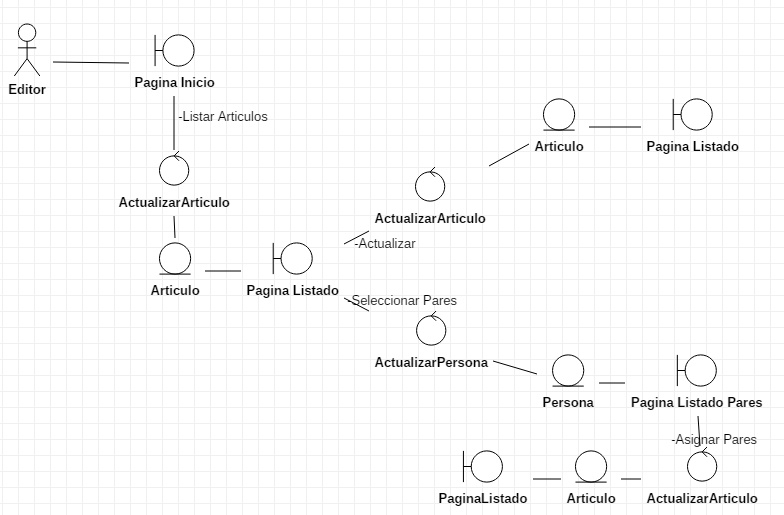
****

**Diagrama de Robustez: Realizar Registro**

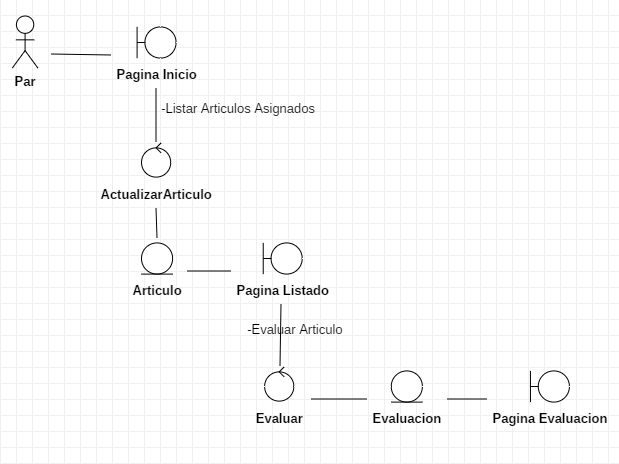
**Diagrama de Robustez: Gestionar autores y pares**

****

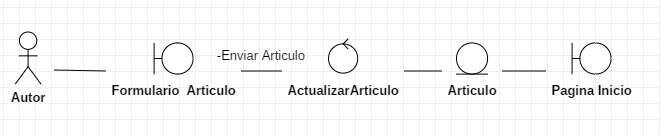
**Diagrama de Robustez: Gestionar artículos**

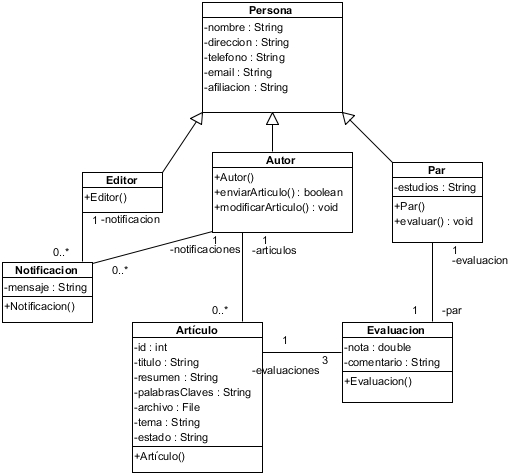
****

**Diagrama de Robustez: Evaluar artículo**

****

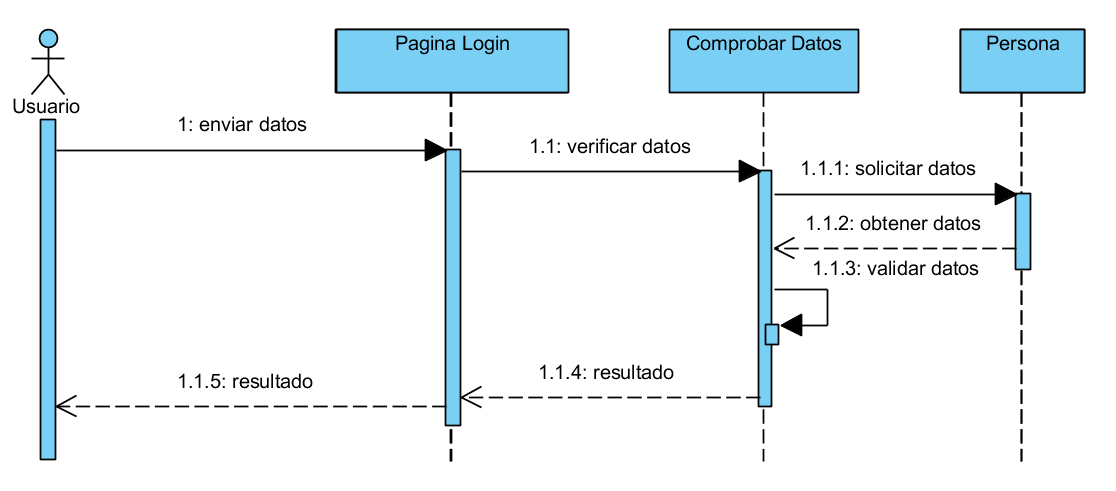
**Diagrama de Robustez: Enviar artículo**

****

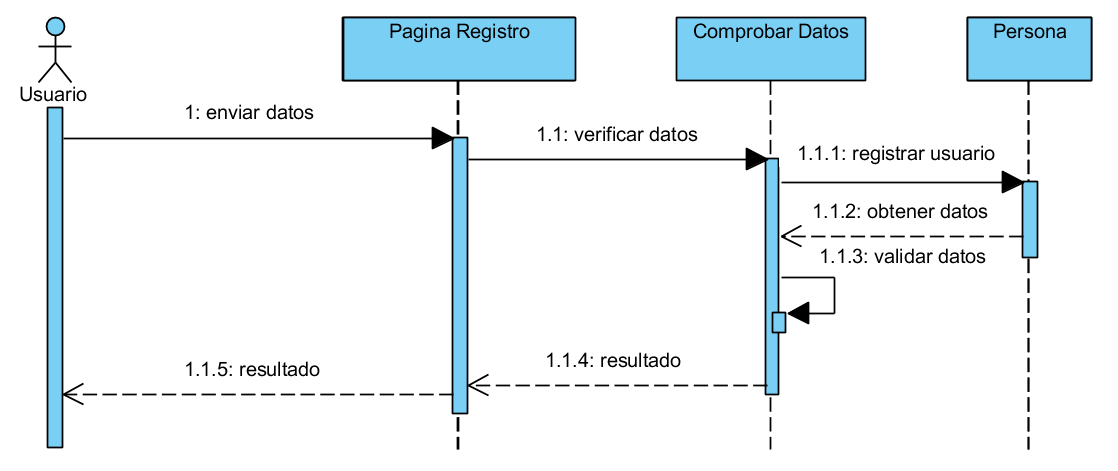
**Diagrama de Clases:**

* + 1. **Diseño**

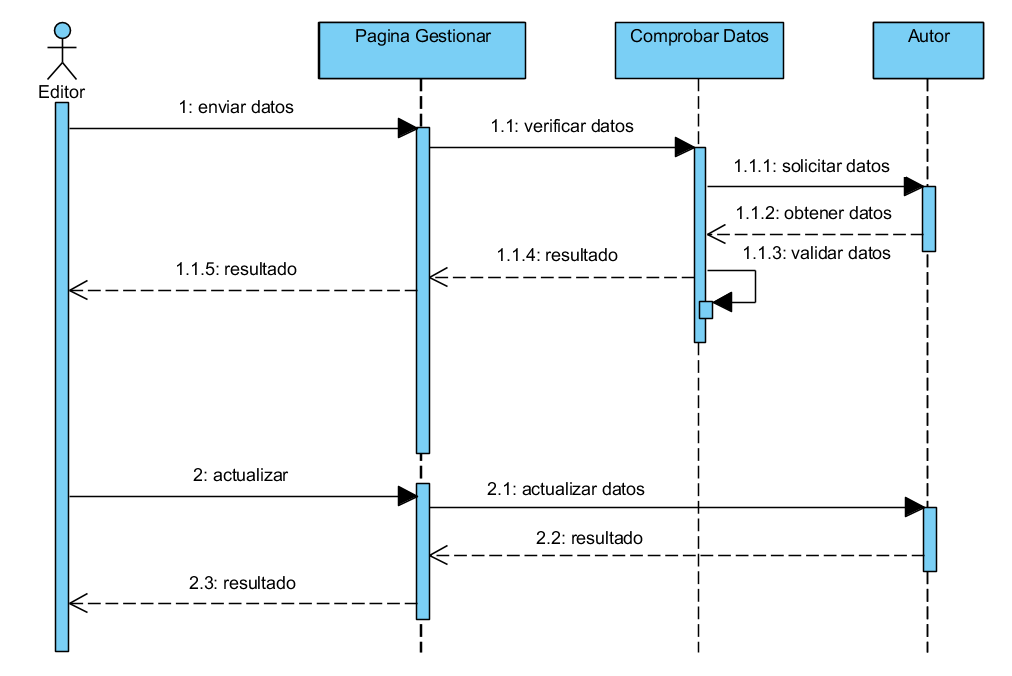
**Diagrama de secuencia: Realizar Login**

****

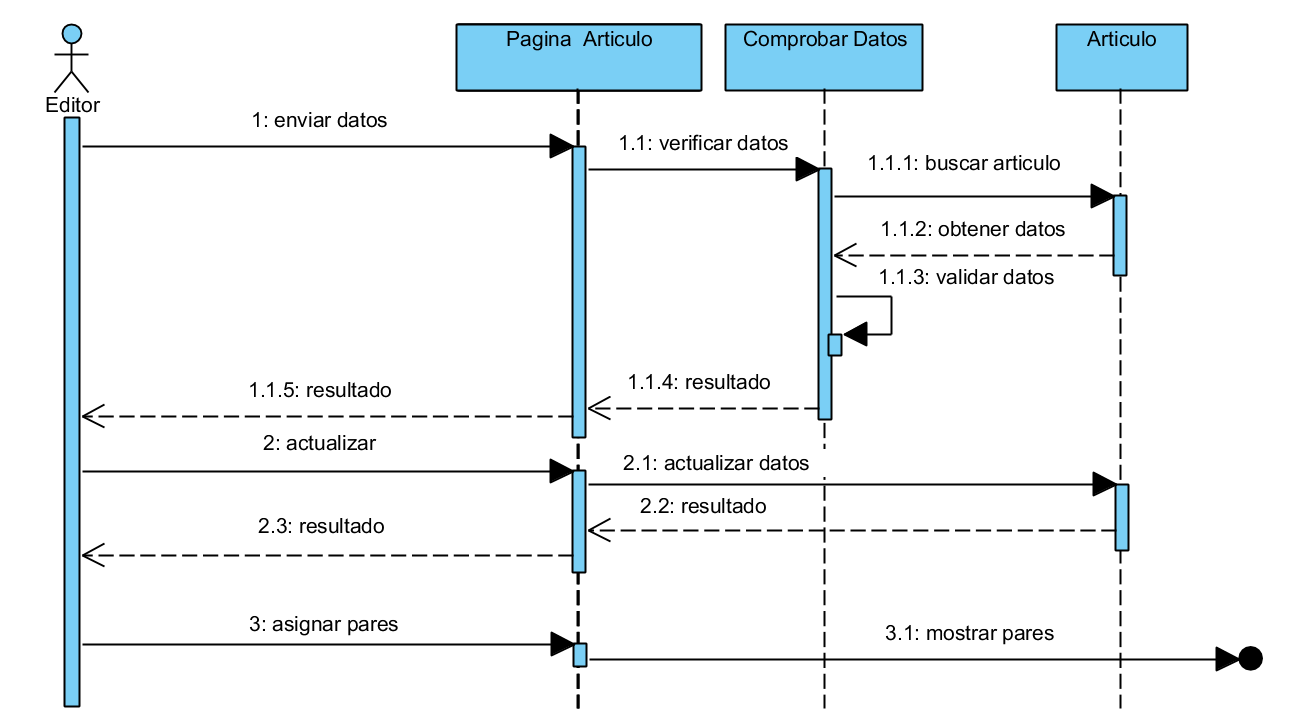
**Diagrama de secuencia: Realizar Registro**



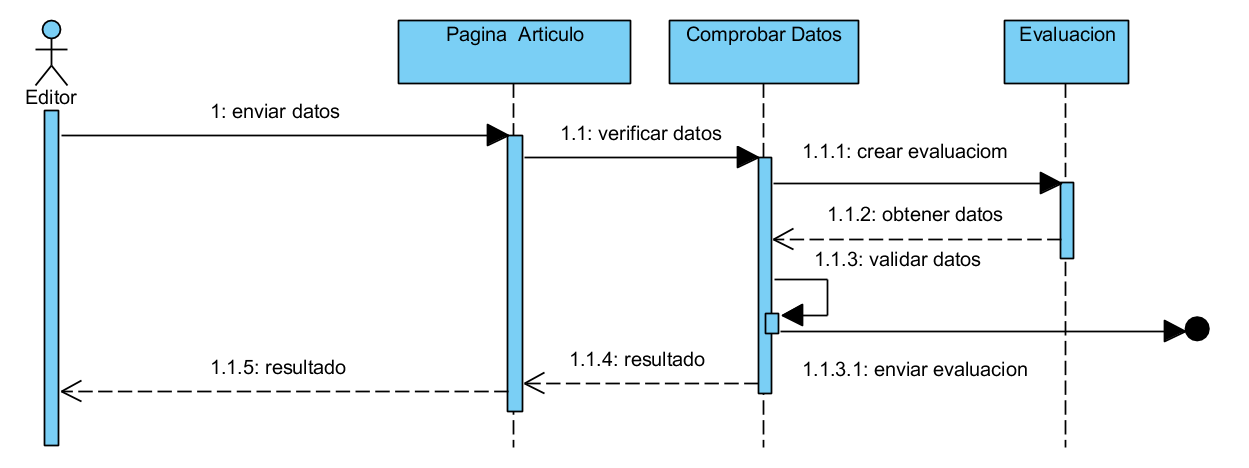
**Diagrama de secuencia: Gestionar autores y pares**



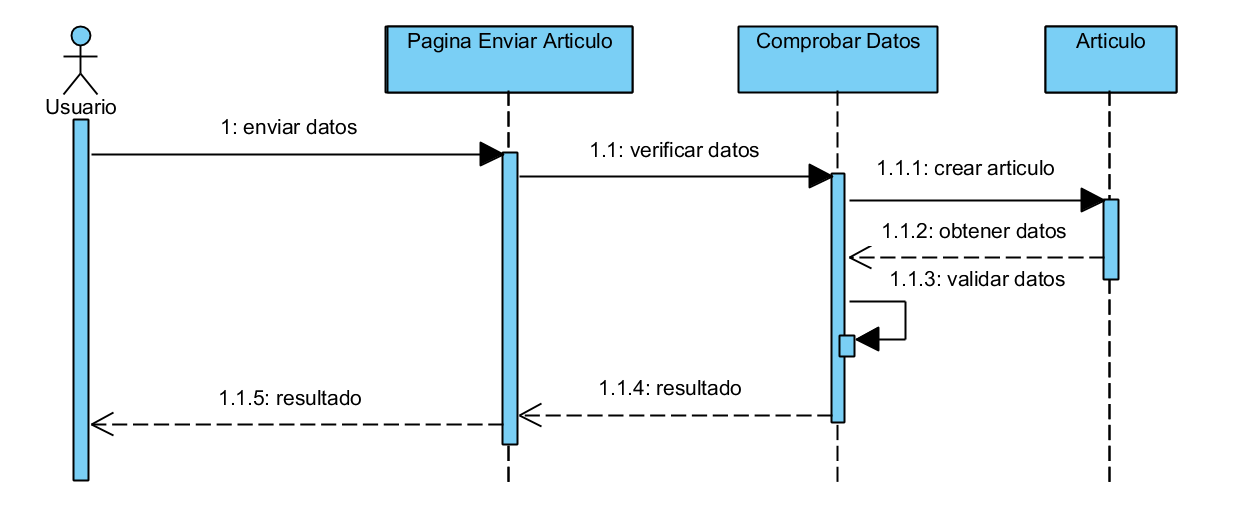
**Diagrama de secuencia: Gestionar artículos**

****

**Diagrama de secuencia: Evaluar artículo**



**Diagrama de secuencia: Enviar artículo**



* + 1. **Implementación**

El desarrollo del proyecto se utilizará una arquitectura Modelo-Vista-Controlador (MVC), que es un patrón de arquitectura de software que separa los datos de la aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. Donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el sistema de gestión de base de datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

La implementación del sistema se realizará en PHP utilizando como framework de desarrollo laravel y las vistas se desarrollarán utilizando HTML, CSS y Boostrap, además se va a utilizar JavaScript para que la pagina sea dinámica y la base de datos que vamos a utilizar es MySql.

**4. Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha Inicio (dd/mm/aa) | Fecha Fin (dd/mm/aa) | Actividad | Responsables |
|
| Primera Entrega | | | |
| 15/09/2018 | 16/09/2018 | Levantamiento de requerimientos | todos |
| 19/09/2018 | 19/09/2018 | Diagrama de casos de uso | Julio, Hector, José |
| 20/09/2018 | 20/09/2018 | Mockups | Daniel |
| 22/09/2018 | 23/09/2018 | Organización de informe | todos |
| 28/09/2018 | 28/09/2018 | Entrega primera entrega | todos |
| Segunda Entrega | | | |
| 13/10/2018 | 14/10/2018 | Definir la metodología que se implementará(iconix) | Todos |
| 13/10/2018 | 14/10/2018 | Corrección casos de uso | Todos |
| 19/10/2018 | 19/10/2018 | Diagrama de modelo de dominio | Todos |
| 20/10/2018 | 21/10/2018 | Diagrama de casos de uso detallados | Todos |
| 26/10/2018 | 26/10/2018 | Diagrama de robustez | Todos |
| 26/10/2018 | 26/10/2018 | Consulta diagramas de robustez | Todos |
| 27/10/2018 | 27/10/2018 | Corrección diagrama de robustez | Todos |
| 27/10/2018 | 27/10/2018 | Organización de informe | Hector, José |
| 27/10/2018 | 27/10/2018 | Actualización diagrama de dominio(diagrama de clases) | Todos |
| 28/10/2018 | 28/10/2018 | Diagrama de secuencia | Todos |
| 28/10/2018 | 28/10/2018 | Mockup página inicio | Daniel |
| 31/10/2018 | 31/10/2018 | Organización de informe | todos |
| Tercera Entrega | | | |
| 10/11/2018 | 10/11/2018 | Inicio implementación Frontend registro y login | Todos |
| 11/11/2018 | 11/11/2018 | implementación Backend registro y login (CRUD persona) | Todos |
| 14/11/2018 | 14/11/2018 | implementación Frontend autor | Todos |
| 16/11/2018 | 16/11/2018 | implementación Backend autor (CRUD artículo) | Todos |
| 17/11/2018 | 17/11/2018 | implementación Frontend Par evaluador | Todos |
| 18/11/2018 | 18/11/2018 | implementación Backend par evaluador | Todos |
| 21/11/2018 | 21/11/2018 | Implementación Frontend editor | Todos |
| 23/11/2018 | 24/11/2018 | Implementación Backend editor | Todos |
| 25/11/2018 | 25/11/2018 | implementación Página principal | Todos |
| 28/11/2018 | 28/11/2018 | Pruebas de funcionamiento | Todos |
| 01/12/2018 | 01/12/2018 | Corrección de errores (si los hay) | Todos |