



UNSA
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERIA, PRODUCCION Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION

WorldCare

INTEGRANTES: ANGEL TOMAS CONCHA LAYME
KATE ITATI OLAZABAL CHAVEZ
ANGELA MARGARITA AQUISE SANTOS
EDWARD LUIS HUAYLLASCO CARLOS
JORGE TITO CCAHUAYA
PROFESOR: EDSON LUQUE MAMANI

FECHA DE REALIZACIÓN: 20 DE JULIO DE 2022
FECHA DE ENTREGA: 20 DE JULIO DE 2022

AREQUIPA-PERU

Índice de Contenidos

1. Introducción	1
1.1. Propósito	1
1.2. Alcance del producto	1
1.3. Recursos Humanos	1
1.3.1. Scrum Master	1
1.3.2. Product Owner	1
1.3.3. Scrum Team	1
1.4. Visión general del producto	2
1.4.1. Perspectiva del producto	2
1.4.2. Funciones del producto	2
1.4.3. Características del usuario y limitaciones	2
1.4.4. Caso de Uso	2
1.5. Lluvia de Ideas	3
1.6. Requisitos específicos	3
1.6.1. Requisitos funcionales	3
1.6.2. Requisitos no funcionales	7
1.7. Mockups	10
1.7.1. Página Principal	11
1.7.2. Publicaciones	12
1.7.3. Login	13
1.7.4. Registro	14
1.7.5. Sección Informativa	15
1.8. Trello	16
1.9. Arquitectura	17
1.9.1. Flask y MVC	17
1.9.2. Capa Modelo	17
1.9.3. Capa Vista	18
1.9.4. Capa Controlador	18

Índice de Figuras

1. Diagrama de Caso de Uso	2
2. Diagrama de Lluvia de ideas	3
3. Página Principal	11
4. Publicaciones de Usuarios	12
5. Login	13
6. Registro	14
7. Sección Informativa	15
8. Herramienta para la gestión y desarrollo del proyecto	16
9. Table User	17
10. Bloques html	18
11. Blueprint-Controlador	18

12.	render template - Controlador	19
-----	---	----

Índice de Tablas

1.	Comparación de características: Marketing	2
2.	Requisito funcional RF-001	3
3.	Requisito funcional RF-002	4
4.	Requisito funcional RF-003	4
5.	Requisito funcional RF-004	4
6.	Requisito funcional RF-005	5
7.	Requisito funcional RF-006	5
8.	Requisito funcional RF-007	5
9.	Requisito funcional RF-008	5
10.	Requisito funcional RF-009	6
11.	Requisito funcional RF-010	6
12.	Requisito funcional RF-011	6
13.	Requisito funcional RF-012	7
14.	Requisito no funcional RNF-001	7
15.	Requisito no funcional RNF-002	7
16.	Requisito no funcional RNF-003	8
17.	Requisito no funcional RNF-004	8
18.	Requisito no funcional RNF-005	8
19.	Requisito no funcional RNF-006	9
20.	Requisito no funcional RNF-007	9
21.	Requisito no funcional RNF-008	9
22.	Requisito no funcional RNF-009	9

1. Introducción

1.1. Propósito

Crear un sistema con propósito social informativo que permita a un usuario realizar cálculos y consultas sobre su huella de carbono, así como descubrir noticias y recomendaciones positivas relacionadas con el cuidado del medio ambiente.

1.2. Alcance del producto

El sistema a desarrollar pretende motivar a los usuarios que lo utilicen, para lograr esto no solo las noticias y tips que se muestran son positivos, sino que la huella de carbono calculada se traducirá un “hecho” positivo en el mundo real.

1.3. Recursos Humanos

La división de cada integrante en un ámbito en específico, para así poder distribuir el trabajo de una manera mucho más fácil y equitativa, utilizando la metodología Scrum, decidimos dar los siguientes cargos a cada integrante:

1.3.1. Scrum Master

El Scrum Master de nuestro equipo es Kate Itati Olazabal Chavez ya que fue quien tuvo la idea sobre este proyecto, ella será la encargada de guiarnos y resolver las dudas que nos puedan surgir en el desarrollo del proyecto.

1.3.2. Product Owner

El Product Owner de nuestro equipo es Angel Concha Layme, él será la voz del cliente, en otras palabras, será el enlace entre un cliente y nuestro equipo de desarrollo. Él lleva la visión del producto y lo que se necesita desarrollar, además de gestionar los comentarios de los usuarios.

1.3.3. Scrum Team

El Scrum Team estará compuesto por Angela Aquisé Santos, Edward Luis Huayllasco Carlos, Jorge Alfredo Tito, quienes serán los encargados de convertir World Care en un software entregable. En este caso decidimos dividir el trabajo de cada uno de nosotros en dos partes.

- **Diseñador** Quien se encargara de generar el diseño arquitectónico y el diseño detallado del sistema será Angela Aquisé Santos, basándose en los requisitos.
- **Programadores** Los encargados son Edward Luis Huayllasco y Jorge Alfredo Tito, convierten los requisitos del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de apoyo.

1.4. Visión general del producto

1.4.1. Perspectiva del producto

El sistema **WorldCare** será un producto diseñado para trabajar en Web, lo que permitirá en fácil acceso y distribución.

1.4.2. Funciones del producto

1.4.3. Características del usuario y limitaciones

Tabla 1: Comparación de características: Marketing

Tipo de usuario	Descripción	Limitaciones
Usuario	Cualquier persona que haga uso del sistema World Care	El usuario solo podrá hacer cálculos y consultas sobre su huella de carbono.

1.4.4. Caso de Uso

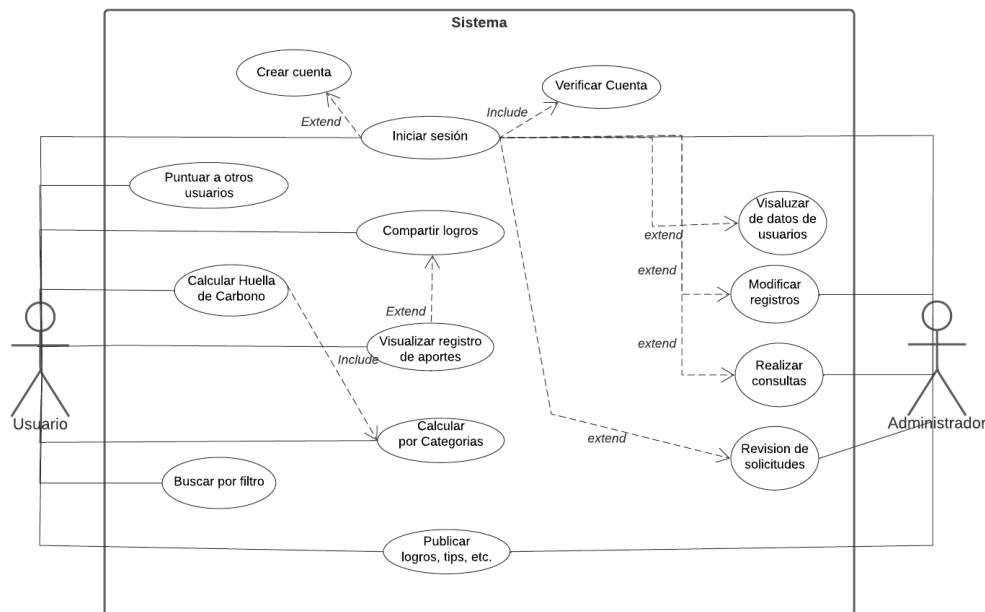


Figura 1: Diagrama de Caso de Uso

1.5. Lluvia de Ideas

Primeras nociones y aportes de cada uno de los integrantes del grupo para tener una idea clara del proyecto a realizarse.

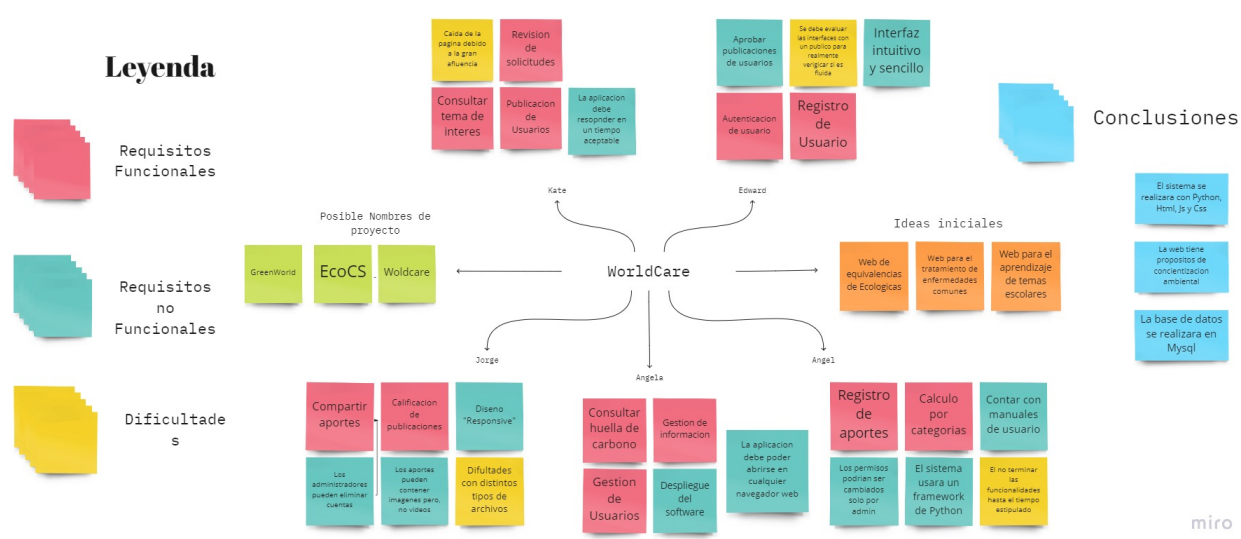


Figura 2: Diagrama de Lluvia de ideas

1.6. Requisitos específicos

1.6.1. Requisitos funcionales

Tabla 2: Requisito funcional RF-001

Identificación del requerimiento	RF-001
Nombre del Requerimiento	Autenticación de Usuario
Descripción del requerimiento	Los usuarios registrados deberán identificarse para poder llevar un perfil de usuario.
Prioridad del requerimiento	Media

Tabla 3: Requisito funcional RF-002

Identificación del requerimiento	RF-002
Nombre del Requerimiento	Registro de Usuario
Descripción del requerimiento	El sistema permitirá a los usuarios registrarse. El usuario debe suministrar datos como: Nombre, Apellido, E-mail, Usuario y Password
Prioridad del requerimiento	Media

Tabla 4: Requisito funcional RF-003

Identificación del requerimiento	RF-003
Nombre del Requerimiento	Consultar tema de interés
Descripción del requerimiento	El sistema permitirá a los usuarios realizar búsquedas relacionadas al tema de su interés
Prioridad del requerimiento	Alta

Tabla 5: Requisito funcional RF-004

Identificación del requerimiento	RF-004
Nombre del Requerimiento	Consultar huella de carbono
Descripción del requerimiento	El sistema permitirá a los usuarios realizar el cálculo de su huella de carbono
Prioridad del requerimiento	Alta

Tabla 6: Requisito funcional RF-005

Identificación del requerimiento	RF-005
Nombre del Requerimiento	Registro de Aportes
Descripción del requerimiento	El sistema permitirá a los usuarios llevar un registro de sus aportes
Prioridad del requerimiento	Alta

Tabla 7: Requisito funcional RF-006

Identificación del requerimiento	RF-006
Nombre del Requerimiento	Compartir Aportes
Descripción del requerimiento	El sistema permitirá a los usuarios compartir sus logros
Prioridad del requerimiento	Alta

Tabla 8: Requisito funcional RF-007

Identificación del requerimiento	RF-007
Nombre del Requerimiento	Cálculo por categorías
Descripción del requerimiento	El sistema permitirá a los usuarios realizar el cálculo de su huella de carbono por categorías
Prioridad del requerimiento	Alta

Tabla 9: Requisito funcional RF-008

Identificación del requerimiento	RF-008
Nombre del Requerimiento	Publicación de Usuarios
Descripción del requerimiento	El sistema permitirá a los usuarios hacer una solicitud para publicar sus logros, ideas, tips, avances, etc.
Prioridad del requerimiento	Media

Tabla 10: Requisito funcional RF-009

Identificación del requerimiento	RF-009
Nombre del Requerimiento	Calificación de publicaciones
Descripción del requerimiento	El sistema permitirá a los usuarios calificar los tips de otros usuarios.
Prioridad del requerimiento	Media

Tabla 11: Requisito funcional RF-010

Identificación del requerimiento	RF-009
Nombre del Requerimiento	Revisión de solicitudes
Descripción del requerimiento	El sistema permitirá al administrador revisar las solicitudes del usuario y aprobarlas o desaprobarlas.
Prioridad del requerimiento	Alta

Tabla 12: Requisito funcional RF-011

Identificación del requerimiento	RF-011
Nombre del Requerimiento	Gestión de información
Descripción del requerimiento	El sistema permitirá al administrador modificar la base de datos, ver registro general, realizar consultas, añadir información, etc.
Prioridad del requerimiento	Alta

Tabla 13: Requisito funcional RF-012

Identificación del requerimiento	RF-012
Nombre del Requerimiento	Gestión de Usuarios
Descripción del requerimiento	El sistema permitirá al administrador visualizar los datos de los usuarios
Prioridad del requerimiento	Alta

1.6.2. Requisitos no funcionales

Tabla 14: Requisito no funcional RNF-001

Identificación del requerimiento	RNF-001
Nombre del Requerimiento	
Descripción del requerimiento	El sistema debe tener una interfaz de uso intuitivo y sencillo
Prioridad del requerimiento	

Tabla 15: Requisito no funcional RNF-002

Identificación del requerimiento	RNF-002
Nombre del Requerimiento	
Descripción del requerimiento	El sistema debe contar con manuales de usuario estructurados adecuadamente.
Prioridad del requerimiento	

Tabla 16: Requisito no funcional RNF-003

Identificación del requerimiento	RNF-003
Nombre del Requerimiento	
Descripción del requerimiento	La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tablets y teléfonos inteligentes.
Prioridad del requerimiento	

Tabla 17: Requisito no funcional RNF-004

Identificación del requerimiento	RNF-004
Nombre del Requerimiento	
Descripción del requerimiento	Toda funcionalidad del sistema debe responder en un tiempo razonable
Prioridad del requerimiento	

Tabla 18: Requisito no funcional RNF-005

Identificación del requerimiento	RNF-005
Nombre del Requerimiento	
Descripción del requerimiento	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
Prioridad del requerimiento	

Tabla 19: Requisito no funcional RNF-006

Identificación del requerimiento	RNF-006
Nombre del Requerimiento	
Descripción del requerimiento	El despliegue del software
Prioridad del requerimiento	

Tabla 20: Requisito no funcional RNF-007

Identificación del requerimiento	RNF-007
Nombre del Requerimiento	
Descripción del requerimiento	Los administradores son responsables y son los unicos que pueden eliminar si asi lo desean una cuenta del servicio Web
Prioridad del requerimiento	

Tabla 21: Requisito no funcional RNF-008

Identificación del requerimiento	RNF-008
Nombre del Requerimiento	
Descripción del requerimiento	Las publicaciones pueden únicamente tener archivos de imágenes adjuntas (PNG o JPG), no se aceptan videos
Prioridad del requerimiento	

Tabla 22: Requisito no funcional RNF-009

Identificación del requerimiento	RNF-009
Nombre del Requerimiento	
Descripción del requerimiento	La aplicación se puede abrir desde cualquier navegador web.
Prioridad del requerimiento	

1.7. Mockups

1.7.1. Página Principal



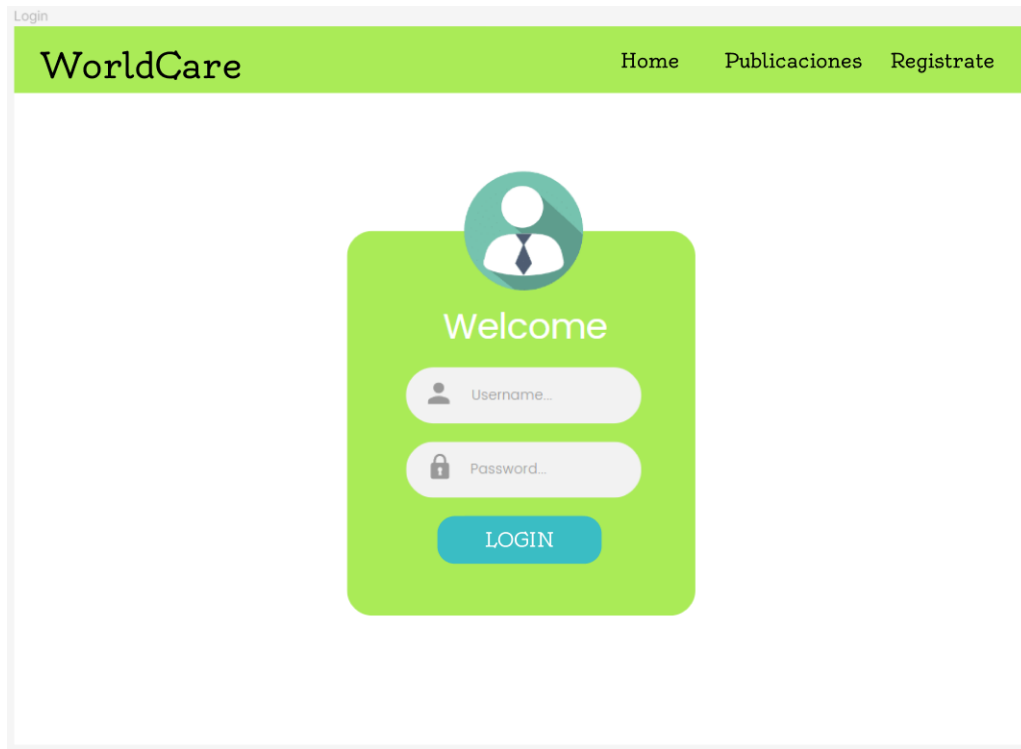
Figura 3: Página Principal

1.7.2. Publicaciones



Figura 4: Publicaciones de Usuarios

1.7.3. Login



The image shows a web browser window with the title "Login". The page has a green header bar with the "WorldCare" logo on the left and navigation links "Home", "Publicaciones", and "Registrate" on the right. The main content area is white and features a central green rounded rectangle. Inside this rectangle, there is a circular profile icon at the top, followed by the word "Welcome" in white. Below "Welcome" are two input fields: "Username..." with a person icon and "Password..." with a lock icon. At the bottom of the green rectangle is a blue button labeled "LOGIN".

Figura 5: Login

1.7.4. Registro



The image shows a web browser window displaying the registration page of WorldCare. The browser's address bar shows "Registrate". The page has a green header with the "WorldCare" logo on the left and navigation links "Home", "Publicaciones", and "Registrate" on the right. The "Registrate" link is underlined. The main content area features a green rounded rectangle with a white user icon at the top. Below the icon, the text "Unete a Nosotros!" is displayed. Underneath this text are five input fields, each with a small icon and a placeholder text: "Nombre...", "Apellidos...", "Correo...", "Username...", and "Password...". The "Password..." field has a lock icon. At the bottom of the green rectangle is a blue button with the text "Comencemos".

Figura 6: Registro

1.7.5. Sección Informativa

Infórmate

WorldCare

Home

Infórmate

Aportes

Publicaciones

Regístrate

El poder está en el conocimiento



Las energías renovables batan (nuevamente) récords

La lucha contra el cambio climático y los compromisos internacionales están siendo clave para acelerar la llamada «transición energética» y, sin duda, es una gran noticia el hecho de que en 2021 se hayan vuelto a romper los records: según datos la Agencia Internacional de Energías Renovables (IRENA), en 2020 la capacidad energética renovable creció un 50% más que en 2019. Y no solo eso: se prevé que para finales de 2026 el 95% de la capacidad mundial de generación de energía provenga de energías renovables.

Ecofira 2022

Feria Internacional de las Soluciones Medioambientales y Transición Ecológica



Figura 7: Sección Informativa

1.8. Trello

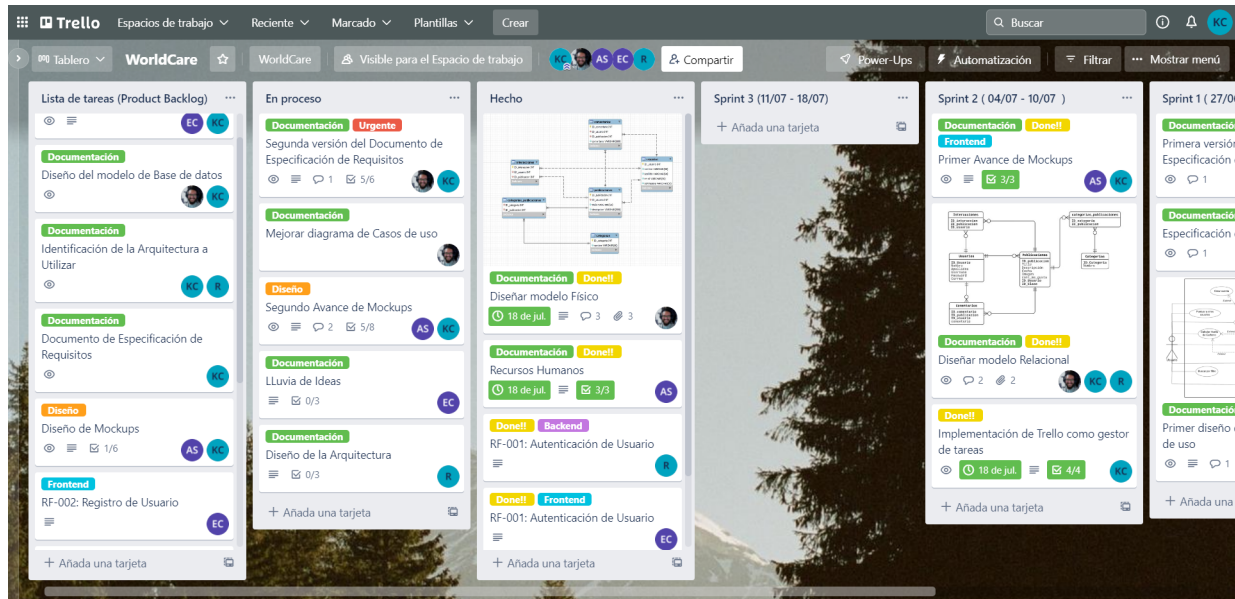


Figura 8: Herramienta para la gestión y desarrollo del proyecto

1.9. Arquitectura

Del Ingles Model View Controller - MVC, el Modelo-vista-controlador, es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

1.9.1. Flask y MVC

- Aunque por defecto no tiene un ORM, podemos usar una extensión de Flask para definir el modelo de datos. Esta característica nos abstrae del uso del motor de Base de Datos y lo hace independiente.
- Con Flask vamos a definir un controlador, que es capaz de determinar las rutas con las que accedemos a la aplicación, procesar la información necesaria y mostrar la información necesaria en cada momento.
- Flask utiliza jinja2 como motor de plantillas, con lo que es muy fácil diseñar las vistas que vamos a mostrar a los usuarios en cada momento.

El patrón de diseño MVC es soportado en Flask se divide en tres capas ya mencionadas:

1.9.2. Capa Modelo

- **SQLAlchemy (9).** SQLAlchemy es el kit de herramientas SQL de Python y el asignador relacional de objetos que brinda a los desarrolladores de aplicaciones todo el poder y la flexibilidad de SQL. Proporciona un conjunto completo de patrones de persistencia de nivel empresarial bien conocidos, diseñados para un acceso a bases de datos eficiente y de alto rendimiento, adaptados a un lenguaje de dominio simple y Pythonic.
- Select, Insert, Delete - Flask-SQLAlchemy).

```
6 class User(db.Model):
7     __tablename__ = 'users'
8     id = db.Column(db.Integer, primary_key=True)
9     username = db.Column(db.String(50))
10    password = db.Column(db.Text)
11    rol = db.Column(db.Integer, db.ForeignKey(
12        'roles.id'), nullable=False, default=2)
13    #roles = db.relationship('Role', secondary='user_roles')
14    roles = relationship(Role, backref=db.backref(
15        "children04", cascade="all,delete"))
```

Figura 9: Table User

1.9.3. Capa Vista

- **Sistema de plantillas Jinja2 (10).** Jinja es un motor de plantillas rápido, expresivo y extensible. Los marcadores de posición especiales en la plantilla permiten escribir código similar a la sintaxis de Python. Luego, se pasan datos a la plantilla para representar el documento final.

```
1  {% extends "base.html" %}
2  {% block header %}
3      <h2> {% block title %}
4          Crear Categoría de publicaciones
5      {% endblock %} </h2>
6  {% endblock %}
7
8  {% block content %}
9      <form action="" method="post">
10         <label for="title">Nombre</label>
11         <input type="text" name="name" id="name" requerid>
12         <input type="submit" value="Guardar">
13     </form>
14
15     <h1>Clases de publicaciones</h1>
16     {% for iter in clasesPub %}
17         <h5>{{iter.nombre}}</h5>
18     {% endfor %}
19 {% endblock %}
```

Figura 10: Bloques html

1.9.4. Capa Controlador

- **Blueprints (11).** Los Blueprints son módulos con los que se construyen las aplicaciones Flask. Los objetos Blueprints son similares a Flask, pero con la diferencia de que una aplicación sólo tendrá un objeto Flask, mientras que puede tener varios Blueprints. La ventaja de su uso es que para aplicaciones largas puedo distribuir el código en varios ficheros, en lugar de tenerlos todo en un único fichero.
- `render template (12).`

```
22
23  blog = Blueprint('blog', __name__, url_prefix='/blog')
24
```

Figura 11: Blueprint-Controlador

```
54 @blog.route("/")
55 def index():
56     updatePostLikes()
57     posts = Post.query.all()
58     posts = list(reversed(posts))
59     interacciones = Interaccion.query.all()
60     db.session.commit()
61     db.session.commit()
62     return render_template('blog/index.html', interacciones=interacciones,
63                             posts=posts, get_user=get_user, functionInter=InteraccionUserIn
64                             getTipoPublicacion=getTipoPublicacion)
65
```

Figura 12: render template - Controlador