

# U4 S14 AVANCE DE PROYECTO

SOF - S - MA - 7 - 1  
APLICACIONES WEB AVANZADO  
ING. CEDENO JUAN CARLOS

## INTEGRANTES:

- GUTIERREZ SUAREZ RONALD JOSEPH
- LECARO PACHECO ODETTE NICOLE
- MARURI VARGAS MAYERLI VIVIANA
- ORDOÑEZ AVILES GENESIS GABRIELA

**Requisitos funcionales:**

Requisito Funcional	Descripción
<b>RF1: Registro de Usuarios</b>	Los usuarios deben poder registrarse proporcionando su nombre, apellido, correo electrónico único, contraseña, fecha de nacimiento y seleccionando una carrera. Además, pueden optar por completar una biografía y cargar una foto de perfil.
<b>RF2: Inicio de Sesión</b>	Los usuarios registrados deben poder iniciar sesión usando su nombre de usuario y contraseña.
<b>RF3: Modificación de Perfil</b>	Los usuarios deben poder modificar su información personal, incluyendo nombre, apellido, biografía, foto de perfil y contraseña.
<b>RF4: Publicación de Contenido</b>	Los usuarios registrados pueden crear publicaciones que contengan texto. Estas publicaciones se relacionan con el usuario que las creó.
<b>RF5: Comentarios en Publicaciones</b>	Los usuarios pueden comentar en las publicaciones de otros usuarios. Cada comentario se relaciona con la publicación y el usuario que lo realiza.
<b>RF6: Reacciones a Publicaciones</b>	Los usuarios pueden reaccionar a las publicaciones con opciones como "me gusta", "me encanta", etc. Cada reacción se asocia con la publicación y el usuario que reacciona.
<b>RF7: Envío de Mensajes Grupales</b>	Los usuarios pueden enviar mensajes en un chat grupal. Cada mensaje se relaciona con el usuario que lo envía.
<b>RF8: Envío de Mensajes Individuales</b>	Los usuarios pueden enviar mensajes privados entre sí. Cada mensaje se asocia con el usuario emisor y el receptor, y tiene una marca de tiempo de envío.

## BASE DE DATOS:

-- Tabla de carreras

```
select * from proyectop2.carreras
```

```
CREATE TABLE proyectop2.carreras (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO proyectop2.carreras (nombre) VALUES
```

```
('Arquitectura'),
```

```
('Ciencia de Datos e Inteligencia Artificial'),
```

```
('Ingeniería Civil'),
```

```
('Ingeniería de los Alimentos'),
```

```
('Ingeniería Industrial'),
```

```
('Ingeniería Ambiental'),
```

```
('Software'),
```

```
('Sistemas de Información'),
```

```
('Tecnologías de la Información'),
```

```
('Telemática'),
```

```
('Agronomía'),
```

```
('Agropecuaria'),
```

```
('Medicina Veterinaria'),
```

```
('Gastronomía'),
```

```
('Biología'),
```

```
('Bioquímica y Farmacia'),
```

```
('Dietética y Nutrición'),
```

```
('Enfermería'),
```

('Fonoaudiología'),  
( 'Odontología'),  
( 'Terapia de Lenguaje'),  
( 'Ciencias de la Educación Mención Informática'),  
( 'Administración de Empresas'),  
( 'Comercio Exterior'),  
( 'Contabilidad y Auditoría'),  
( 'Economía'),  
( 'Economía Internacional'),  
( 'Finanzas'),  
( 'Gestión de la Información Gerencial'),  
( 'Mercadotecnia'),  
( 'Negocios Internacionales'),  
( 'Turismo'),  
( 'Otro');

```
select * from proyectop2.usuarios
```

-- Tabla de usuarios

```
CREATE TABLE proyectop2.usuarios (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    apellido VARCHAR(100) NOT NULL,  
    correo VARCHAR(100) UNIQUE NOT NULL,  
    contraseña VARCHAR(255) NOT NULL,  
    fecha_nacimiento DATE,  
    biografia TEXT,  
    foto_perfil VARCHAR(255),
```

```
id_carrera INTEGER REFERENCES proyectop2.carreras(id)
);
```

```
ALTER TABLE proyectop2.usuarios
ADD COLUMN user_state BOOLEAN DEFAULT true;
```

```
ALTER TABLE proyectop2.usuarios
ADD COLUMN usuario VARCHAR(15);
```

```
ALTER TABLE proyectop2.usuarios
ADD CONSTRAINT unique_usuario UNIQUE (usuario);
```

```
select * from proyectop2.publicaciones
```

-- Tabla de publicaciones

```
CREATE TABLE proyectop2.publicaciones (
    id SERIAL PRIMARY KEY,
    id_usuario INTEGER REFERENCES proyectop2.usuarios(id),
    contenido TEXT NOT NULL
);
```

-- Tabla de comentarios

```
CREATE TABLE proyectop2.comentarios (
    id SERIAL PRIMARY KEY,
    id_publicacion INTEGER REFERENCES proyectop2.publicaciones(id),
    id_usuario INTEGER REFERENCES proyectop2.usuarios(id),
    contenido TEXT NOT NULL
);
```

-- Tabla de reacciones

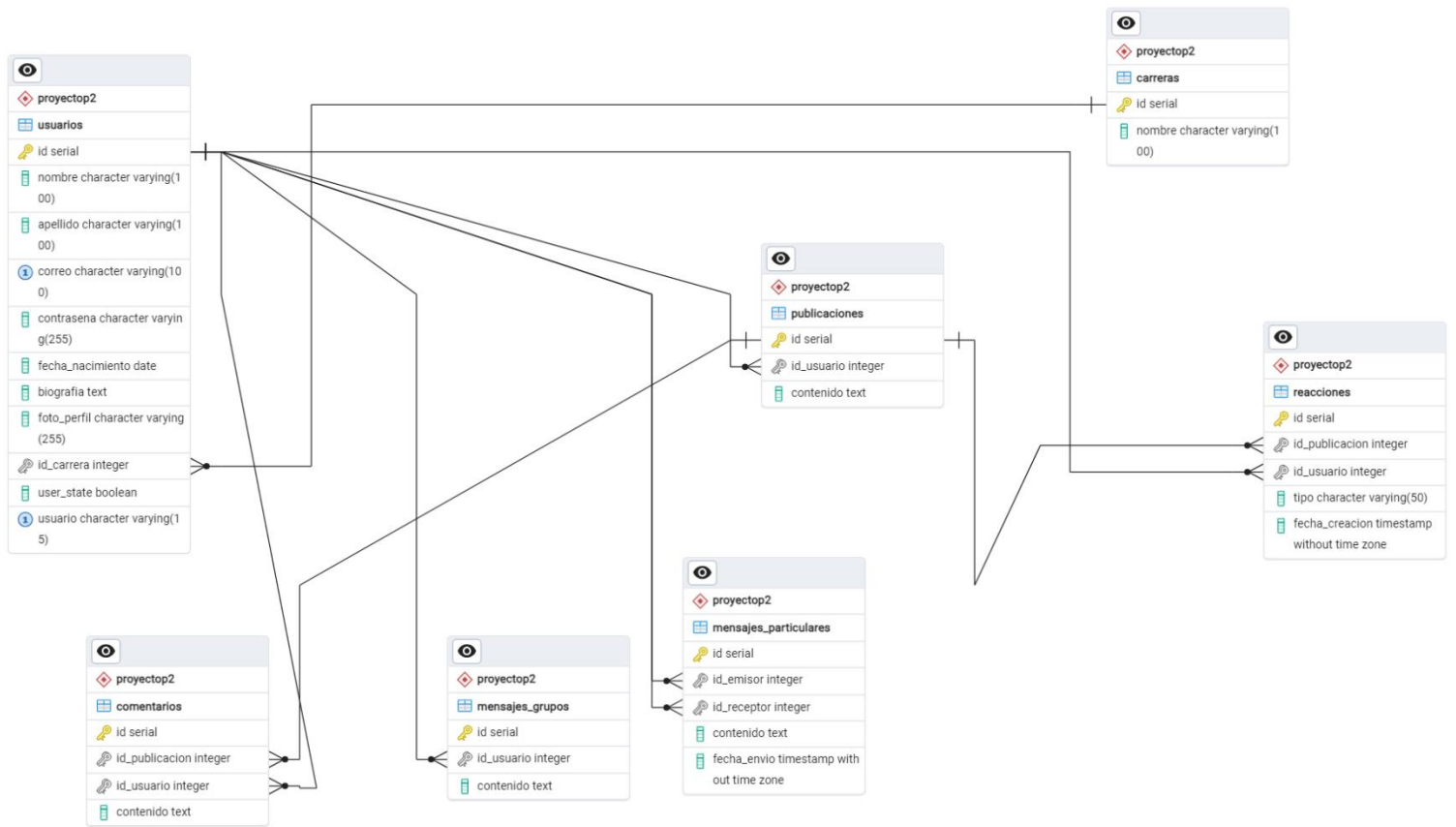
```
CREATE TABLE proyectop2.reacciones (  
    id SERIAL PRIMARY KEY,  
    id_publicacion INTEGER REFERENCES proyectop2.publicaciones(id),  
    id_usuario INTEGER REFERENCES proyectop2.usuarios(id),  
    tipo VARCHAR(50) NOT NULL CHECK (tipo IN ('me gusta', 'me encanta', 'me  
divierte', 'me sorprende', 'me entristece', 'me enfada')),  
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

-- Tabla de mensajes general

```
CREATE TABLE proyectop2.mensajes_grupos (  
    id SERIAL PRIMARY KEY,  
    id_usuario INTEGER REFERENCES proyectop2.usuarios(id) ON DELETE  
CASCADE,  
    contenido TEXT NOT NULL  
);
```

-- Tabla de mensajes individuales o particulares entre usuarios

```
CREATE TABLE proyectop2.mensajes_particulares (  
    id SERIAL PRIMARY KEY,  
    id_emisor INTEGER REFERENCES proyectop2.usuarios(id) ON DELETE  
CASCADE,  
    id_receptor INTEGER REFERENCES proyectop2.usuarios(id) ON DELETE  
CASCADE,  
    contenido TEXT NOT NULL,  
    fecha_envio TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```



## BACKEND:

```

src
├── api
│   └── Components
│       ├── ActualizarComentarioComponent.py
│       ├── ActualizarPublicacionComponent.py
│       ├── ActualizarUsuarioComponent.py
│       ├── AmistadActualizarComponent.py
│       ├── ChatInsertarComponent.py
│       ├── ContarReaccionComponent.py
│       ├── creacionamistadComponent.py
│       ├── CreacionComentarioComponent.py
│       ├── CreacionGrupoComponent.py
│       ├── CreacionHistoriaComponent.py
│       ├── creacionMiembrosGrupo_Component.py
│       ├── creacionpubli_component.py
│       ├── CreacionPubliGrupoComponent.py
│       ├── CreacionReaccionComponent.py
│       ├── creacionusuario_component.py
│       ├── datosUsuarios.py
│       ├── EliminacionPublicacionComponent.py
│       ├── EliminarComentarioComponent.py
│       ├── EliminarUsuarioComponent.py
│       ├── HistorialMensajesGrupoCOM.py
│       ├── list_carreras_component.py
│       ├── ListaAceptadoAmigoComponent.py
│       ├── ListaComentariosComponent.py
│       ├── ListaGruposComponents.py
│       ├── ListaPublicacionesCarreraComponent.py
│       ├── ListaPublicacionUsuarioComponent.py

```

```

ListaAceptadoAmigoComponent.py
ListaComentariosComponent.py
ListaGruposComponents.py
ListaPublicacionesCarreraComponent.py
ListaPublicacionUsuarioComponent.py
ListaUsuarioByldCompon.py
ListaUsuariosComponent.py
listpubli_components.py
login_component.py
MensajeCarreraComponent.py
ObtenerNombreComponent.py
ReaccionActualizarComponent.py
├── Model
│   ├── Request
│   │   └── login_request.py
│   └── Response
├── Routes
│   └── routes.py
└── Services
    ├── ActualizarComentarioService.py
    ├── ActualizarPublicacionServie.py
    ├── ActualizaUsuarioService.py
    ├── AmistadActualizarService.py
    ├── carreras_service.py
    ├── ChatInsertarService.py
    ├── ComentarioCreationService.py
    ├── ContarReaccionServie.py
    ├── Creacion_Reaccion_service.py
    └── creacion_usuario.py

```

- creacionamistadService.py
- CreacionMiembroGruoService.py
- creacionpubli\_servic.py
- CreacionPubliGrupoService.py
- EliminarComentarioService.py
- EliminarPublicacionService.py
- GrupoCreationService.py
- HistoriaCreationService.py
- HistorialChatGrupSERV.py
- ListaAmigosAceptado.py
- ListaComentarioPublicacion.py
- ListaGrupoService.py
- ListaUsuarioByIdService.py
- ListaUsuarioService.py
- listPubli\_service.py
- login\_service.py
- MensajesPorCarreraService.py
- ObtenerNombreService.py
- PublicacionesPorCarreraService.py
- PublicacionUsuarioService.py
- ReaccionActualizarService.py
- usuario\_service.py
- UsuarioEliminarService.py

## Reacciones Operaciones relacionadas con reacciones

**PUT** /reaccion/actualizar Actualizar Reacción

**POST** /reaccion/create Crear Reacción

**GET** /reaccion/conteo Contar Reacciones

## Amistades Operaciones relacionadas con amistades

**GET** /usuarios/aceptados/{user\_id} Listar Amigos Aceptados

**PUT** /amistad/update Actualizar Estado de Amistad

**POST** /amistad/create Crear Amistad

## Usuarios Operaciones relacionadas con usuarios

**PUT** /usuario/actualizar Actualizar Usuario

**PUT** /usuario/actualizarEstado Actualizar Estado del Usuario es decir eliminar

**POST** /usuarios/create Crear Usuario

**GET** /usuario/list Listar Usuarios

**GET** /usuario/id/{username} Obtener ID de usuario por nombre de usuario

**GET** /api/usuarios/{user\_id} Obtener el nombre de un usuario por su ID

**GET** /api/Litsausuarios/{user\_id} Obtener Usuario por ID

## Carreras Operaciones relacionadas con carreras

**GET** /carreras/List Listar todas las carreras



Carreras	Operaciones relacionadas con carreras	▼
GET	/carreras/List	Listar todas las carreras
Publicaciones	Operaciones relacionadas con publicaciones	▼
POST	/publicacion/create	Crear Publicación
GET	/list/publicaciones	Listar Publicaciones
PUT	/publicacion/update	Actualizar Publicación
Comentarios Publicaciones	Operaciones relacionadas con comentarios en publicaciones	▼
POST	/comentario/create	Crear Comentario
PUT	/comentario/update	Actualizar Comentarios
DELETE	/comentario/{id}/delete	Eliminar Comentarios
Grupos	Operaciones relacionadas con grupos	▼
POST	/grupo_miembro/agregar	Agregar Miembro a Grupo
GET	/list/grupos	Listar Grupos
POST	/grupo/create	Crear Grupo
POST	/publicacion/grupo/create	Crear Publicación en Grupo
Historias	Operaciones relacionadas con historias	▼
POST	/historia/create	Crear Historia
Comentarios		▼
GET	/api/comentarios/{post_id}	Listar Comentarios por Publicación
Mensajes		▼
POST	/mensajes/agregar	Agregar Mensaje al Grupo
GET	/mensajes/{id_grupo}	Obtener Mensajes del Grupo
GET	/mensajes/carrera/{user_id}	Obtener Mensajes de la Carrera del Usuario
WS Security		▼
POST	/security/login	Login de Usuario

FRONTEND:



¡Bienvenido a **UnIEDU**!

Una red social que podrás interactuar con otros estudiantes de tu carrera o otras que tengas curiosidad aprender.

## ¿Por qué un buho nos representa?



**UNIEDU**

*Un búho como personaje principal para la red social estudiantil porque simboliza sabiduría y conocimiento, valores clave para nuestra plataforma. Su imagen amistosa y sabia refleja el espíritu de aprendizaje y conexión entre estudiantes, ofreciendo una experiencia acogedora y estimulante.*

INICIAR SESIÓN

REGISTRARSE

Nombre de Usuario \*

Contraseña \*

INGRESAR

[¿Olvidaste tu contraseña?](#)



¡Bienvenido a **UnIEDU**!

Una red social que podrás interactuar con otros estudiantes de tu carrera o otras que tengas curiosidad aprender.

## ¿Por qué un buho nos representa?



**UNIEDU**

INICIAR SESIÓN

REGISTRARSE

Nombre \*

Apellido \*

Correo Electrónico \*

Nombre de Usuario \*

Contraseña \*

Fecha de Nacimiento \*

dd/mm/aaaa



Bioografía \*



g

Bienvenido/a **genesisoa!** (ID: 1)



### Crear Publicación

Escribe algo interesante de tu carrera? Ven y comparte con los demás...

PUBLICAR

### Publicaciones generales



Genesis Gabriela

Hola mi primera publicacion :D



Genesis Gabriela

Segunda publicacion



Genesis Gabriela

Tercera Publicacion

### Aquí están las publicaciones sobre tu carrera!



Genesis Gabriela

...

### Grupo de chat general de tu carrera

Personas de tu carrera, te pueden ayudar con dudas!

Odette

...

IMBOX

### Tercera Publicacion

### Aquí están las publicaciones sobre tu carrera!



Genesis Gabriela

...

Hola mi primera publicacion :D

😊 0 😊 0 😊 1 😊 0 😊 0 😊 0 😊 0

[VER COMENTARIOS](#)



Genesis Gabriela

...

Segunda publicacion

😊 1 😊 0 😊 0 😊 0 😊 0 😊 0 😊 0

[VER COMENTARIOS](#)



Genesis Gabriela

...

### Grupo de chat general de tu carrera



Hola

Usuario: Genesis Gabriela



hola

Usuario: Genesis Gabriela

Nuevo mensaje



Personas de tu carrera, te pueden ayudar con dudas!

Odette

Carrera:  
Arquitectura

IMBOX