CSHUB

Integrantes:

-Eduardo Medina
-Jorge Rebosio

El proyecto / Introducción

- Somos una comunidad diferente

- Nuestros temas de conversación son distintos.

- Necesitamos un espacio que nos una como carrera y nos permita conectarnos como personas.

Solución

CSHUB

Entities / User

```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(70), index=True, unique=True)
    email = db.Column(db.String(90), index=True, unique=True)
    password_hash = db.Column(db.String(110))
    posts = db.relationship('Post', backref='author', lazy='dynamic')
    gender = db.Column(db.String(5))
    about_me = db.Column(db.String(140))
    last_seen = db.Column(db.DateTime, default=datetime.utcnow)
    followed = db.relationship(
        'User', secondary=followers,
        primaryjoin=(followers.c.follower_id == id),
        secondaryjoin=(followers.c.followed_id == id),
        backref=db.backref('followers', lazy='dynamic'), lazy='dynamic')
    def __repr__(self):
        return '<User {}>'.format(self.username)

    def set_password(self, password):
        self.password_hash = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password_hash, password)
```

Entities / Cursos - Posts

```python
class Curso(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(90), index = True, unique = True)
    profesor_id = db.Column(db.Integer, db.ForeignKey('profesor.id'))

    def __repr__(self):
        return '<Curso {}>'.format(self.name)

class Post(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    body = db.Column(db.String(140))
    timestamp = db.Column(db.DateTime, index=True, default=datetime.utcnow)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'))

    def __repr__(self):
        return '<Post {}>'.format(self.body)
```

Código / Registro

```python
@app.route('/register', methods=['GET', 'POST'] )
def register():
    if current_user.is_authenticated:
        return redirect(url_for('index'))
    form = RegistrationForm()
    if form.validate_on_submit():
        user = User(username=form.username.data, email=form.email.data, gender=form.gender.data)
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        flash('Felicitaciones! Ahora perteneces a la comunidad de Cientificos de la Computacion en UTEC.')
        return redirect(url_for('login'))
    return render_template('register.html', title='Register', form=form)
```

Código / logIn

```python
@app.route('/login', methods=['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user is None or not user.check_password(form.password.data):
            flash('Nombre de usuario o password equivocado.')
            return redirect(url_for('login'))
        login_user(user, remember=form.remember_me.data)
    return render_template('login.html', title='Sing In', form=form)
```

Código / LogOut

```
@app.route('/logout')
def logout():
    logout_user()
    return redirect(url_for('index'))
```

CÓDIGO/Editar Perfil

```python
@app.route('/edit_profile', methods=['GET', 'POST'])
@login_required
def edit_profile():
    form = EditProfileForm(current_user.username)
    if form.validate_on_submit():
        current_user.username = form.username.data
        current_user.about_me = form.about_me.data
        db.session.commit()
        flash('Los cambios han sido guardados.')
        return redirect(url_for('edit_profile'))
    elif request.method == 'GET':
        form.username.data = current_user.username
        form.about_me.data = current_user.about_me
    return render_template('edit_profile.html', title='Edit Profile', form=form)
```

Código/Explorar

```
@app.route('/explore')
@login_required
def explore():
    posts = Post.query.order_by(Post.timestamp.desc()).all()
    return render_template('index.html', title='Explore', posts=posts)
```

Código/Delete user and posts

```python
@app.route('/delete/<username>/')
@login_required
def delete(username):
    user = User.query.filter_by(username=username).first()
    for post in user.posts:
        user.posts.remove(post)
    db.session.delete(user)
    db.session.commit()
    flash('Usuario eliminado')
    return render_template("index.html", title='Deleted')
```