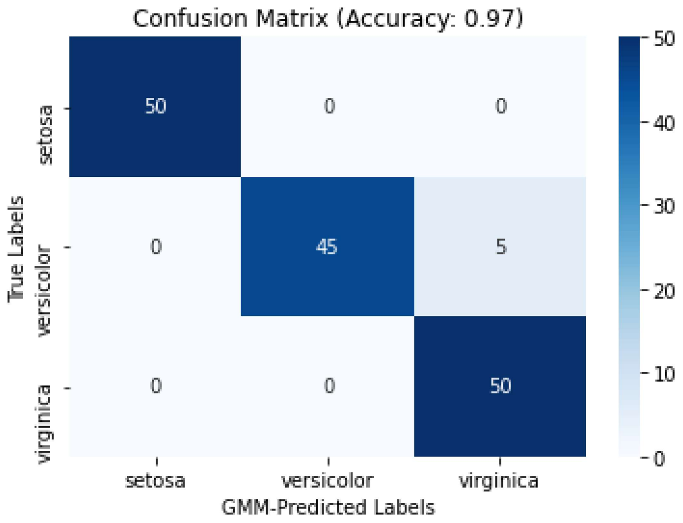
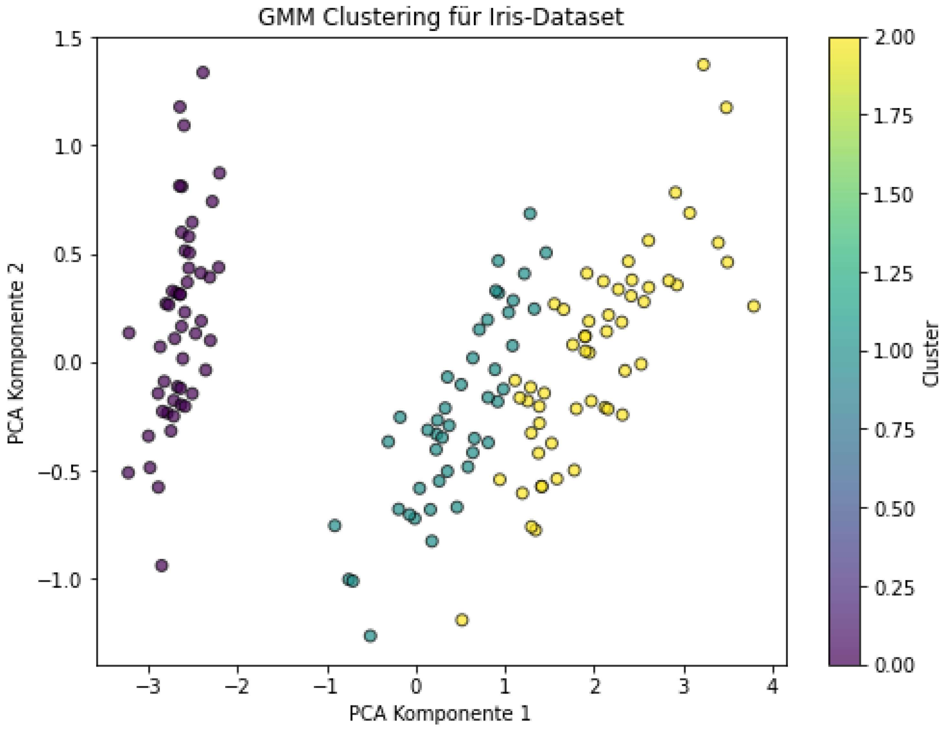
In [1]:

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.mixture import GaussianMixture
from sklearn.decomposition import PCA
from sklearn.metrics import confusion_matrix, accuracy_score
import seaborn as sns

# Iris-Daten laden
iris = datasets.load_iris()
X = iris.data  # Nur Merkmale
y_true = iris.target  # Tatsächliche Labels

# GMM mit 3 Clustern erstellen
gmm = GaussianMixture(n_components=3, random_state=42)
gmm.fit(X)
y_pred = gmm.predict(X)

# Cluster-Labels anpassen (GMM-Labels stimmen nicht unbedingt mit den echten überein)
from scipy.stats import mode
def map_labels(y_true, y_pred):
    labels = np.zeros_like(y_pred)
    for i in np.unique(y_pred):
        mask = (y_pred == i)
        labels[mask] = mode(y_true[mask])[0]
    return labels

y_mapped = map_labels(y_true, y_pred)

# Confusion Matrix berechnen
conf_matrix = confusion_matrix(y_true, y_mapped)
acc = accuracy_score(y_true, y_mapped)

# Confusion Matrix plotten
plt.figure(figsize=(6,4))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel("GMM-Predicted Labels")
plt.ylabel("True Labels")
plt.title(f"Confusion Matrix (Accuracy: {acc:.2f})")
plt.show()

# PCA für 2D-Visualisierung
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# Visualisierung der Cluster
plt.figure(figsize=(8,6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y_mapped, cmap='viridis', edgecolors='k', alpha=0.7)
plt.title("GMM Clustering für Iris-Dataset")
plt.xlabel("PCA Komponente 1")
plt.ylabel("PCA Komponente 2")
plt.colorbar(label="Cluster")
plt.show()
```


Confusion Matrix (Accuracy: 0.97)
True Labels
setosa
versicolor
virginica
GMM-Predicted Labels
setosa
versicolor
virginica
50
0
0
0
45
5
0
0
50
50
40
30
20
10
0


GMM Clustering für Iris-Dataset
PCA Komponente 2
PCA Komponente 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−3
−2
−1
0
1
2
3
4
Cluster
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00