



Projecte missatgeria. Shared preferences.

A l'hora de fer la validació convé guardar el codi d'usuari i el token que ens retorna el servidor REST com a resposta a un login correcta.

També ens convé guardar el nom d'usuari i la contrasenya correctes per tal de que cada vegada que l'aplicació s'iniciï abans de demanar les credencials a l'usuari intentem la validació amb la parella usuari/password guardada. D'aquesta manera estalviam a l'usuari la feina de validar-se manualment cada vegada.

Per guardar aquesta informació el millor es fer servir la classe SharedPreferences (també ho podríem guardar dins la base de dades, un fitxer pla,...)

Podríem implementar una classe per tal de gestionar aquesta feina:

```
public class Preferencies {
    private static final String NOM_PREFERENCIES = "PreferenciesQuepassaEh";
    private final static String CLAU_CODIUSUARI="id";
    private final static String CLAU_USER="user";
    private final static String CLAU_PASSWD="passwd";
    private final static String TOKEN="token";
    private int codiusuari;
    private String user;
    private String password;
    private String token;
    private SharedPreferences prefs;
    public Preferencies(Context ctx) {
        this.prefs = ctx.getSharedPreferences(NOM_PREFERENCIES,
ctx.MODE_PRIVATE);
        this.codiusuari = this.prefs.getInt(CLAU_CODIUSUARI, -1);
        this.user = this.prefs.getString(CLAU_USER, "");
        this.password = this.prefs.getString(CLAU_PASSWD, "");
        this.token = this.prefs.getString(TOKEN, "");
    }
    public int getCodiusuari() {
        return codiusuari;
    }
    public void setCodiusuari(int idempleat) {
        this.codiusuari = idempleat;
        SharedPreferences.Editor editor = this.prefs.edit();
        editor.putInt(CLAU_CODIUSUARI, idempleat);
        editor.commit();
    }
    public String getUser() {
        return user;
    }
    public void setUser(String user) {
        this.user = user;
        SharedPreferences.Editor editor = this.prefs.edit();
        editor.putString(CLAU_USER, user);
        editor.commit();
    }
    public String getPassword() {
        return password;
    }
}
```

```
public void setPassword(String password) {
    this.password = password;
    SharedPreferences.Editor editor = this.prefs.edit();
    editor.putString(CLAU_PASSWD,password);
    editor.commit();
}

public String getToken() {
    return token;
}
public void setToken(String token) {
    this.token = token;
    SharedPreferences.Editor editor = this.prefs.edit();
    editor.putString(TOKEN,token);
    editor.commit();
}
}
```