Sistema FIEP

JORNADA DE APRENDIZAGEM

GERÊNCIA DA QUALIDADE

Case: Portal de Gestão de Acessos

KNAPP SUDAMERICA & MAIS INTELIGENCIA

Profª Ana Christina Vanali

Turma 6º/7º Período / 2025

1 Introduction

In a partnership among Sistema FIEP, KNAPP SUDAMERICA and MAIS INTELIGÊNCIA, the proposal to prepare a demand to FIEP where the challenge is related to develop a Portal called MAIS Cloud User Management Portal where the users can sign up and, after being approved by approval/management user, they can access the features inside the profile that had been created for.

1.1 Objective

Develop a web application that serves as a login portal and facilitates the creation of users in Oracle Cloud Infrastructure (OCI) through API requests. The portal should allow user registration, role-based access control, company management, and JSON request generation compatible with OCI.

1.2 Main Goals

- Portal Website for user sign up
- Persist logins into Oracle Database (can be Autonomous Free)
- Create necessary tables to store the:
    - Users
    - Permissions
    - Roles
    - Company
    - Oracle OCI json formats to interact with OCI

2 Requirements

2.1 *Authentication & User Management*

Implement a login page where users can sign in.

Users can self-register but require admin approval before accessing the system.

Maintain role-based access control with the following roles:

Admin: Full access, including user and company management.

Manager: Can manage users within their assigned company.

Operator: Can trigger JSON requests but has no administrative privileges.

The initial login to start to manage the server can be admin/admin

2.2 Database & Data Management

Store user information, login details, and access roles.

Maintain logs of user actions.

Configure necessary tables that will ser as tables of parameters for API communication with OCI.

Store company details such as name and location.

2.3 Company Management

Admin can create and manage company records.

Users are associated with a single company (1:1 relationship).

A company can have multiple users (1:n relationship).

Admin can assign users to a specific company.

2.4 JSON Request Handling & OCI Interaction

Provide a page where users can generate a JSON request to create a user in OCI.

- Ensure the JSON format is compatible with OCI API specifications.
- Provide a button to trigger the JSON request to OCI.

- Maintain logs of triggered requests.

For a web-based platform, you'll most likely use API Key Authentication. This requires:

- Tenancy OCID
- User OCID (an admin user in your tenancy)
- Fingerprint (from your public key)
- Private key (for signing API requests)

API Docs: https://docs.oracle.com/en-us/iaas/api/#/en/identity/

Example 1: Create a User
Endpoint: POST
JSON Payload:
json

```
{
   "compartmentId": "ocid1.tenancy.oc1.maisinteligencia.infrastructure",
   "name": "newuser@dominio-user.com",
   "description": "User for web platform access"
}
```

Example 2: Create a Group
Endpoint: POST
json

```
{
   "compartmentId": "ocid1.tenancy.oc1.maisinteligencia.infrastructure",
   "name": "web-platform-admins",
   "description": "Group for web platform admins"
}
```

Example 3: Create Policy Permission

Endpoint: POST

```
{
  "compartmentId": "ocid1.tenancy.oc1.maisinteligencia.infrastructure",
  "name": "web-platform-policy",
  "description": "Policy to grant web platform permissions",
  "statements": [
    "Allow group web-platform-admins to manage all-resources in tenancy"
  ]
}
```

2.5 CSS Stile

Consider the colors and style as following:

1. --primary-color: rgb(238, 238, 238);
2. --secondary-color: #fcfcfc;
3. --third-color: #e9e9e9;
4. --bg-color: #fefefe;
5. --bg-color-nav: #d2d3d5;
6. --shadow-primary: rgba(0, 0, 0, 0.2);
7. --shadow-secondary: #373a3d;
8. --text-color: black;
9. --text-color-nav: #49494B;
10. --fixed-color: black;
11. --cl-primary: #AEC455;
12. --cl-secondary: #416d9c;
13. --cl-secondary-dark: #345a82;
14. --border-color: rgb(191, 191, 191);
15. --border-secondary: rgba(255, 255, 255, 0.7);

3 UI/UX Considerations

Implement a clean and responsive UI.

Use a dashboard to present relevant information based on user roles.

Include status indicators for pending approvals and request execution.

4 Additional Considerations

- Secure the application using encryption and best security practices.
- Implement API authentication when interacting with OCI.
- Ensure proper validation and error handling for all inputs and actions.

5 Evaluation Criteria

5.1 Functionality

Does the application meet all listed requirements?

5.2 Security

Are best security practices implemented?

5.3 Code Quality

Is the code well-structured and maintainable?

5.4 User Experience

Is the UI intuitive and easy to use?

5.5 OCI Compatibility

Does the JSON request match OCI specification?