

Security

```
classDiagram
    class JwtUtil {
        - properties: SupabaseProperties
        - secretKey: Key
        + init(): void
        + extractAllClaims(token: String): Claims
        + isValid(token: String): boolean
        + extractUsername(token: String): String
        + extractUserRole(token: String): String
        + extractCustomClaims(token: String): Map<String, Object>
    }
    class SecurityFilter {
        - jwtUtil: JwtUtil
        - userRoleService: UserRoleService
        + doFilter(request: ServletRequest, response: ServletResponse, chain: FilterChain): void
    }
    class AuthController {
        - authService: SupabaseAuthService
        + signUp(authRequest: AuthRequest): ResponseEntity<String>
        + signIn(authRequest: AuthRequest): ResponseEntity<String>
    }
    class SupabaseAuthService {
        - http: HttpClient
        - properties: SupabaseProperties
        - mapper: ObjectMapper
        + signUp(email: String, password: String): String
        + signIn(email: String, password: String): String
    }
    class UserRoleService {
        - repo: UserRoleRepository
        + getRolesForUser(userId: UUID): List<String>
    }
    class UserRoleRepository {
        <<interfaz>>
        + findByUserId(userId: UUID): List<UserRole>
    }
    class AppRole {
        <<enumeration>>
        ADMIN
        DECANO
        DIRECTOR_DE_PROGRAMA
        DOCENTE
        COMITE_DE_PROGRAMA
    }
    class UserRole {
        - id: Long
        - userId: UUID
        - role: AppRole
        + getId(): Long
        + setId(id: Long): void
        + getUserId(): UUID
        + setUserId(userId: UUID): void
        + getRole(): AppRole
        + setRole(role: AppRole): void
    }
    JwtUtil -- SecurityFilter
    SecurityFilter -- UserRoleService
    AuthController -- SupabaseAuthService
    SupabaseAuthService -- UserRoleService
    UserRoleService -- UserRoleRepository
    UserRoleRepository -- UserRole
    AppRole -- UserRole
```

The diagram illustrates the security module's architecture. It features several key classes and their interactions:

- JwtUtil**: A utility class for JWT operations, including token validation and claim extraction.
- SecurityFilter**: A filter that utilizes **JwtUtil** and **UserRoleService** to process requests.
- AuthController**: The main entry point for authentication, delegating tasks to **SupabaseAuthService**.
- SupabaseAuthService**: A service that handles authentication logic, interacting with **SupabaseProperties** and **ObjectMapper**.
- UserRoleService**: A service that manages user roles, interacting with **UserRoleRepository**.
- UserRoleRepository**: An interface defining the methods for retrieving user roles.
- AppRole**: An enumeration representing different roles (ADMIN, DECANO, DIRECTOR_DE_PROGRAMA, DOCENTE, COMITE_DE_PROGRAMA).
- UserRole**: A domain entity representing a user's role, with attributes like **id**, **userId**, and **role**.

[illegible]

Entity

```

classDiagram
    class CourseRequirement {
        -id: CourseRequirementId
        -course: Course
        -program: Program
        -prerequisiteCourse: Course
        +getId(): CourseRequirementId
        +setId(id: CourseRequirementId): void
        +getCourse(): Course
        +setCourse(course: Course): void
        +getProgram(): Program
        +setProgram(program: Program): void
        +getPrerequisiteCourse(): Course
        +setPrerequisiteCourse(prerequisiteCourse: Course): void
    }
    class CourseRequirementId {
        -courseId: Long
        -programId: Long
        -prerequisiteCourseId: Long
        +getCourseId(): Long
        +setCourseId(courseId: Long): void
        +getProgramId(): Long
        +setProgramId(programId: Long): void
        +getPrerequisiteCourseId(): Long
        +setPrerequisiteCourseId(prerequisiteCourseId: Long): void
    }
    class StudyPlanEntry {
        -id: StudyPlanEntryId
        -name: String
        -type: String
        -credits: Integer
        -relation: String
        -area: String
        -cycle: String
        -semester: Integer
        -requirements: Long[]
        +getId(): StudyPlanEntryId
        +getName(): String
        +getType(): String
        +getCredits(): Integer
        +getRelation(): String
        +getArea(): String
        +getCycle(): String
        +getSemester(): Integer
        +getRequirements(): Long[]
    }
    class StudyPlanEntryId {
        -programId: Long
        -courseId: Long
        +getProgramId(): Long
        +getCourseId(): Long
    }
    class Program {
        -id: Long
        -name: String
        -perfilProfesional: String
        -perfilOcupacional: String
        -perfilIngreso: String
        -competencias: String
        -resultadosAprendizajeField: Long
        -duration: Int
        -awardingDegree: String
        +getId(): Long
        +setId(id: Long): void
        +getName(): String
        +setName(name: String): void
        +getPerfilProfesional(): String
        +setPerfilProfesional(perfilProfesional: String): void
        +getPerfilOcupacional(): String
        +setPerfilOcupacional(perfilOcupacional: String): void
        +getPerfilIngreso(): String
        +setPerfilIngreso(perfilIngreso: String): void
        +getCompetencias(): String
        +setCompetencias(competencias: String): void
        +getResultadosAprendizajeField(): Long
        +setResultadosAprendizajeField(resultadosAprendizajeField: Long): void
        +getDuration(): Integer
        +setDuration(duration: Integer): void
        +getAwardingDegree(): String
        +setAwardingDegree(awardingDegree: String): void
    }
    class CourseProgramId {
        -courseId: Long
        -programId: Long
        +getCourseId(): Long
        +setCourseId(courseId: Long): void
        +getProgramId(): Long
        +setProgramId(programId: Long): void
    }
    class CourseProgram {
        -id: CourseProgramId
        -semester: Integer
        -course: Course
        -program: Program
        +getId(): CourseProgramId
        +setId(id: CourseProgramId): void
        +getSemester(): Integer
        +setSemester(semester: Integer): void
        +getCourse(): Course
        +setCourse(course: Course): void
        +getProgram(): Program
        +setProgram(program: Program): void
    }
    class Course {
        -id: Long
        -createdAt: LocalDateTime
        -name: String
        -type: CourseType
        -credits: Integer
        -relation: String
        -area: CourseArea
        -cycle: CourseCycle
        +getId(): Long
        +setId(id: Long): void
        +getCreatedAt(): LocalDateTime
        +setCreatedAt(createdAt: LocalDateTime): void
        +getName(): String
        +setName(name: String): void
        +getType(): CourseType
        +setType(type: CourseType): void
        +getCredits(): Integer
        +setCredits(credits: Integer): void
        +getRelation(): String
        +setRelation(relation: String): void
        +getArea(): CourseArea
        +setArea(area: CourseArea): void
        +getCycle(): CourseCycle
        +setCycle(cycle: CourseCycle): void
    }
    class CourseType {
        <<enumeration>>
        -code: String
        -label: String
        +getCode(): String
        +getLabel(): String
        +fromCode(code: String): CourseType
        +toString(): String
        teorico, practico, teorico_practico
    }
    class CourseArea {
        <<enumeration>>
        -code: String
        -label: String
        +getCode(): String
        +getLabel(): String
        +fromCode(code: String): CourseArea
        +toString(): String
        profesional, basica, complementaria, profundizacion, investigacion
    }
    class CourseCycle {
        <<enumeration>>
        -code: String
        -label: String
        +getCode(): String
        +getLabel(): String
        +fromCode(code: String): CourseCycle
        +toString(): String
        acceso_especifico, fundamentacion, no_aplica
    }
    CourseRequirement <|-- CourseProgram
    StudyPlanEntry <|-- StudyPlanEntryId
    CourseRequirement --> CourseRequirementId
    StudyPlanEntry --> StudyPlanEntryId
    Program --> CourseRequirement
    Program --> StudyPlanEntry
    CourseProgram --> CourseRequirement
    CourseProgram --> CourseProgramId
    CourseProgram --> Course
    CourseProgram --> CourseType
    CourseProgram --> CourseArea
    CourseProgram --> CourseCycle
    Course --> CourseType
    Course --> CourseArea
    Course --> CourseCycle
    
```

The diagram illustrates the following entities and their relationships:

- CourseRequirement** (Base Class):
 - Attributes: `-id: CourseRequirementId`, `-course: Course`, `-program: Program`, `-prerequisiteCourse: Course`
 - Methods: `+getId(): CourseRequirementId`, `+setId(id: CourseRequirementId): void`, `+getCourse(): Course`, `+setCourse(course: Course): void`, `+getProgram(): Program`, `+setProgram(program: Program): void`, `+getPrerequisiteCourse(): Course`, `+setPrerequisiteCourse(prerequisiteCourse: Course): void`
- CourseRequirementId** (Derived Class):
 - Attributes: `-courseId: Long`, `-programId: Long`, `-prerequisiteCourseId: Long`
 - Methods: `+getCourseId(): Long`, `+setCourseId(courseId: Long): void`, `+getProgramId(): Long`, `+setProgramId(programId: Long): void`, `+getPrerequisiteCourseId(): Long`, `+setPrerequisiteCourseId(prerequisiteCourseId: Long): void`
- StudyPlanEntry** (Base Class):
 - Attributes: `-id: StudyPlanEntryId`, `-name: String`, `-type: String`, `-credits: Integer`, `-relation: String`, `-area: String`, `-cycle: String`, `-semester: Integer`, `-requirements: Long[]`
 - Methods: `+getId(): StudyPlanEntryId`, `+getName(): String`, `+getType(): String`, `+getCredits(): Integer`, `+getRelation(): String`, `+getArea(): String`, `+getCycle(): String`, `+getSemester(): Integer`, `+getRequirements(): Long[]`
- StudyPlanEntryId** (Derived Class):
 - Attributes: `-programId: Long`, `-courseId: Long`
 - Methods: `+getProgramId(): Long`, `+getCourseId(): Long`
- Program**:
 - Attributes: `-id: Long`, `-name: String`, `-perfilProfesional: String`, `-perfilOcupacional: String`, `-perfilIngreso: String`, `-competencias: String`, `-resultadosAprendizajeField: Long`, `-duration: Int`, `-awardingDegree: String`
 - Methods: `+getId(): Long`, `+setId(id: Long): void`, `+getName(): String`, `+setName(name: String): void`, `+getPerfilProfesional(): String`, `+setPerfilProfesional(perfilProfesional: String): void`, `+getPerfilOcupacional(): String`, `+setPerfilOcupacional(perfilOcupacional: String): void`, `+getPerfilIngreso(): String`, `+setPerfilIngreso(perfilIngreso: String): void`, `+getCompetencias(): String`, `+setCompetencias(competencias: String): void`, `+getResultadosAprendizajeField(): Long`, `+setResultadosAprendizajeField(resultadosAprendizajeField: Long): void`, `+getDuration(): Integer`, `+setDuration(duration: Integer): void`, `+getAwardingDegree(): String`, `+setAwardingDegree(awardingDegree: String): void`
- CourseProgramId** (Derived Class):
 - Attributes: `-courseId: Long`, `-programId: Long`
 - Methods: `+getCourseId(): Long`, `+setCourseId(courseId: Long): void`, `+getProgramId(): Long`, `+setProgramId(programId: Long): void`
- CourseProgram** (Derived Class):
 - Attributes: `-id: CourseProgramId`, `-semester: Integer`, `-course: Course`, `-program: Program`
 - Methods: `+getId(): CourseProgramId`, `+setId(id: CourseProgramId): void`, `+getSemester(): Integer`, `+setSemester(semester: Integer): void`, `+getCourse(): Course`, `+setCourse(course: Course): void`, `+getProgram(): Program`, `+setProgram(program: Program): void`
- Course**:
 - Attributes: `-id: Long`, `-createdAt: LocalDateTime`, `-name: String`, `-type: CourseType`, `-credits: Integer`, `-relation: String`, `-area: CourseArea`, `-cycle: CourseCycle`
 - Methods: `+getId(): Long`, `+setId(id: Long): void`, `+getCreatedAt(): LocalDateTime`, `+setCreatedAt(createdAt: LocalDateTime): void`, `+getName(): String`, `+setName(name: String): void`, `+getType(): CourseType`, `+setType(type: CourseType): void`, `+getCredits(): Integer`, `+setCredits(credits: Integer): void`, `+getRelation(): String`, `+setRelation(relation: String): void`, `+getArea(): CourseArea`, `+setArea(area: CourseArea): void`, `+getCycle(): CourseCycle`, `+setCycle(cycle: CourseCycle): void`
- CourseType** (Enumeration):
 - Attributes: `-code: String`, `-label: String`
 - Methods: `+getCode(): String`, `+getLabel(): String`, `+fromCode(code: String): CourseType`, `+toString(): String`
 - Values: `teorico`, `practico`, `teorico_practico`
- CourseArea** (Enumeration):
 - Attributes: `-code: String`, `-label: String`
 - Methods: `+getCode(): String`, `+getLabel(): String`, `+fromCode(code: String): CourseArea`, `+toString(): String`
 - Values: `profesional`, `basica`, `complementaria`, `profundizacion`, `investigacion`
- CourseCycle** (Enumeration):
 - Attributes: `-code: String`, `-label: String`
 - Methods: `+getCode(): String`, `+getLabel(): String`, `+fromCode(code: String): CourseCycle`, `+toString(): String`
 - Values: `acceso_especifico`, `fundamentacion`, `no_aplica`

Relationships:

- Generalization:** `CourseRequirement` is the base class for `CourseProgram`. `StudyPlanEntry` is the base class for `StudyPlanEntryId`.
- Association:** `CourseRequirement` is associated with `CourseRequirementId`. `StudyPlanEntry` is associated with `StudyPlanEntryId`.
- Association:** `Program` is associated with `CourseRequirement` and `StudyPlanEntry`.
- Association:** `CourseProgram` is associated with `CourseRequirement`, `CourseProgramId`, `Course`, `CourseType`, `CourseArea`, and `CourseCycle`.
- Association:** `Course` is associated with `CourseType`, `CourseArea`, and `CourseCycle`.