<<abstract>> Curso **Professor** + nome: String + nome: String + codCurso: Int 0..2 Matricula + sobrenome: String + quantMaxAluno: Int + professorTitular: ProfessorTitular + tempoCasa: Int + aluno: Aluno + codProfessor: Int + professorAdjunto: ProfessorAdjunto + curso: Curso + alunosMatriculados: MutableList< Aluno> + data: Date + Professor(nome: String, sobrenome: String, + Curso(nome: String, codCurso: Int, tempoCasa: Int, codProfessor: Int) + Matricula(aluno: Aluno, curso: Curso, quantMaxAluno: Int) + equals(other: Any): Boolean data: Date) + equals(other: Any): Boolean + adicionarUmAluno(umAluno: Aluno): Boolean + excluirAluno(umAluno: Aluno) 0..* 1 ProfessorTitular **ProfessorAdjunto** Aluno + especialidades: String + quantHoras: Int + nome: String + sobrenome: String + ProfessorTitular(especialidades: String) + ProfessorAdjunto(quantHoras: Int) + codAluno: Int + Aluno(nome: String, sobrenome: String, codAluno: Int) 0..* 0..* + equals(other: Any): Boolean DigitalHouseManager + listaAlunos: MutableList<Aluno> + listaProfessores: MutableList<Professor> 0..* + listaCursos: MutableList<Curso> + listaMatricula: MutableList<Matricula> 0..* + registrarCurso(nome: String, codigoCurso: Int, quantidadeMaximaDeAlunos: Int) + excluirCurso(codigoCurso: Int) + registrarProfessorAdjunto(nome: String, sobrenome codigoProfessor: Int, quatidadeDeHoras: Int) + registrarProfessorTitular(nome: String, sobrenome: codigoProfessor: Int, especialidade: String) + excluirProfessor(codigoProfessor: Int) + matricularAluno(nome: String, sobrenome: String, c + excluirAlunoCurso(codigoAluno: Int) + matricularAlunoCurso(codigoAluno: Int, codigoCurs + alocarProfessores(codigoCurso: Int, codigoProfess + visualizarInf(codigoAluno: Int)