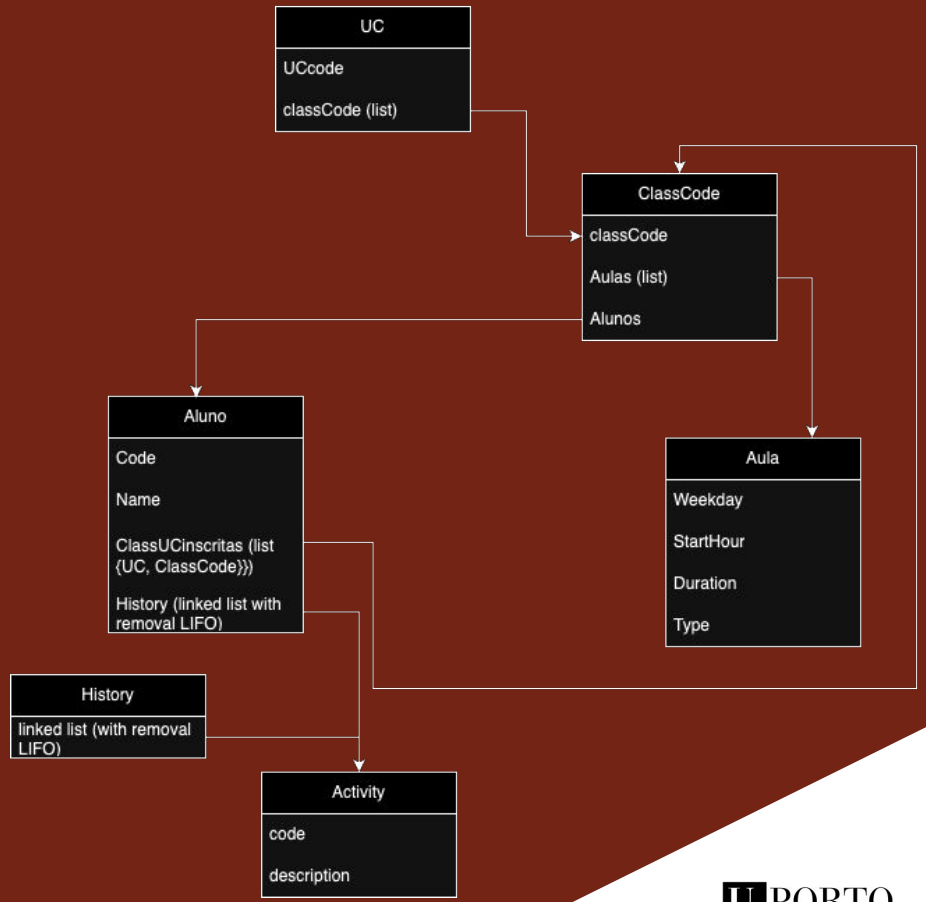
Sistema de Gestão de Horários

Filipe Correia
Gonçalo Nunes
Rodrigo Miranda
Grupo G32

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Inicialmente...



U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Funcionamento

Do prompt de login (estudante vs administrador) a pedidos de mudança de turma.

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Funcionalidades

- Efetuar pedido
  - Mudar de turma numa UC
  - Sair de uma UC
  - Entrar numa turma numa nova UC
- Verificar o horário de:
  - Estudante
  - Turma numa UC
- Listar os estudantes
  - Todos
  - Turma
  - UC
- Número de estudantes registados em N UCs
- Ocupação
  - Turma
  - UC
  - Ano
- Consultar a UC com mais alunos
- Aceitar pedidos
- Verificar último pedido aceite ou pendente
- Desfazer alterações
- Adicionar um novo estudante
- Guardar todas as alterações num ficheiro
- Carregar alterações anteriores

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Exemplo de Utilização

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Estruturas de Dados Usadas

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Student, Uc, ClassCode, Class, Activity

```
class Student {
int studentCode;
std::string name;
std::list<std::pair<Uc&,ClassCode&>> classes;
}
```

```
class ClassCode {
std::string classCode;
std::list<Class> classes;
std::list<int> students;
}
```

```
class Class {
char weekday;
char type;
float startHour;
float duration;
}
```

```
class Uc {
private:
std::string ucCode;
std::list<ClassCode> classes;
}
```

```
class Activity {
int code;
int student;
std::string old;
std::string current;
Uc* uc;
}
```


U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

History, AllUcs, AllStudents

```
class History {
std::stack<Activity> history;
std::queue<Activity> requests;
}
```

```
class AllUcs {
std::list<Uc> ucs;
}
```

```
class AllStudents {
std::set<Student>
students;
}
```


U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Uni

```
class Uni {
        AllUcs ucs;
        AllStudents students;
        bool isAdmin, loggedIn = false;
        int student_id_loggedin = 0;
        History history;
}
```

Complexidade Ciclomática Temporal

void AllStudents::removeStudent(int student): O(log(n))

void AllStudents::changeClassStudent(int student, ClassCode& oldclass, ClassCode& newclass, Uc& uc): O(log(n))

bool AllUcs::ucExists(std::string ucCode): O(n)

void ClassCode::getClasses(std::list<std::pair<const Class&, std::string>>& allClasses, std::string ucCode) const : O(n)

Métodos em History: O(1)

AllStudents parse_students(AllUcs& ucs): Best case: O(n)

int Uc::minOcupation(): O(n)

bool Uni::doesNotColide(int studentCode, const ClassCode& exit, const ClassCode& enter) const: O(n)

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO