

S.A.S.A.

T.P.I., Laboratorio de Programación, Semestre 1°, 2010

INDEX

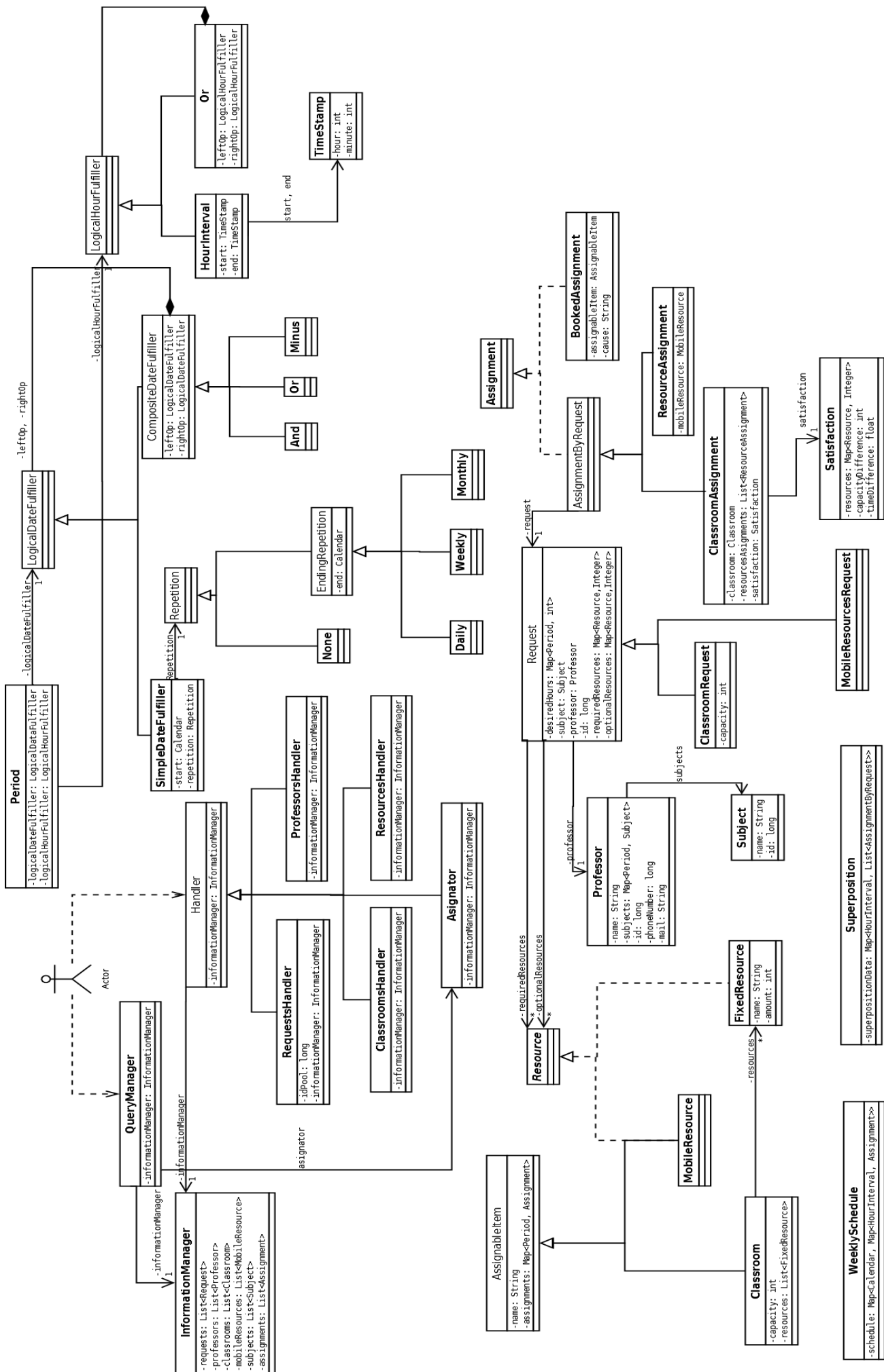
Brief explanation	3
U.M.L. class diagram	4
U.M.L. class diagram details:logical time/date	5
U.M.L. class diagram details: Managers, Handlers and User interaction	6
U.M.L. class diagram details: Unit structures	7
Classes and methods	8
Workgroup divisions	12

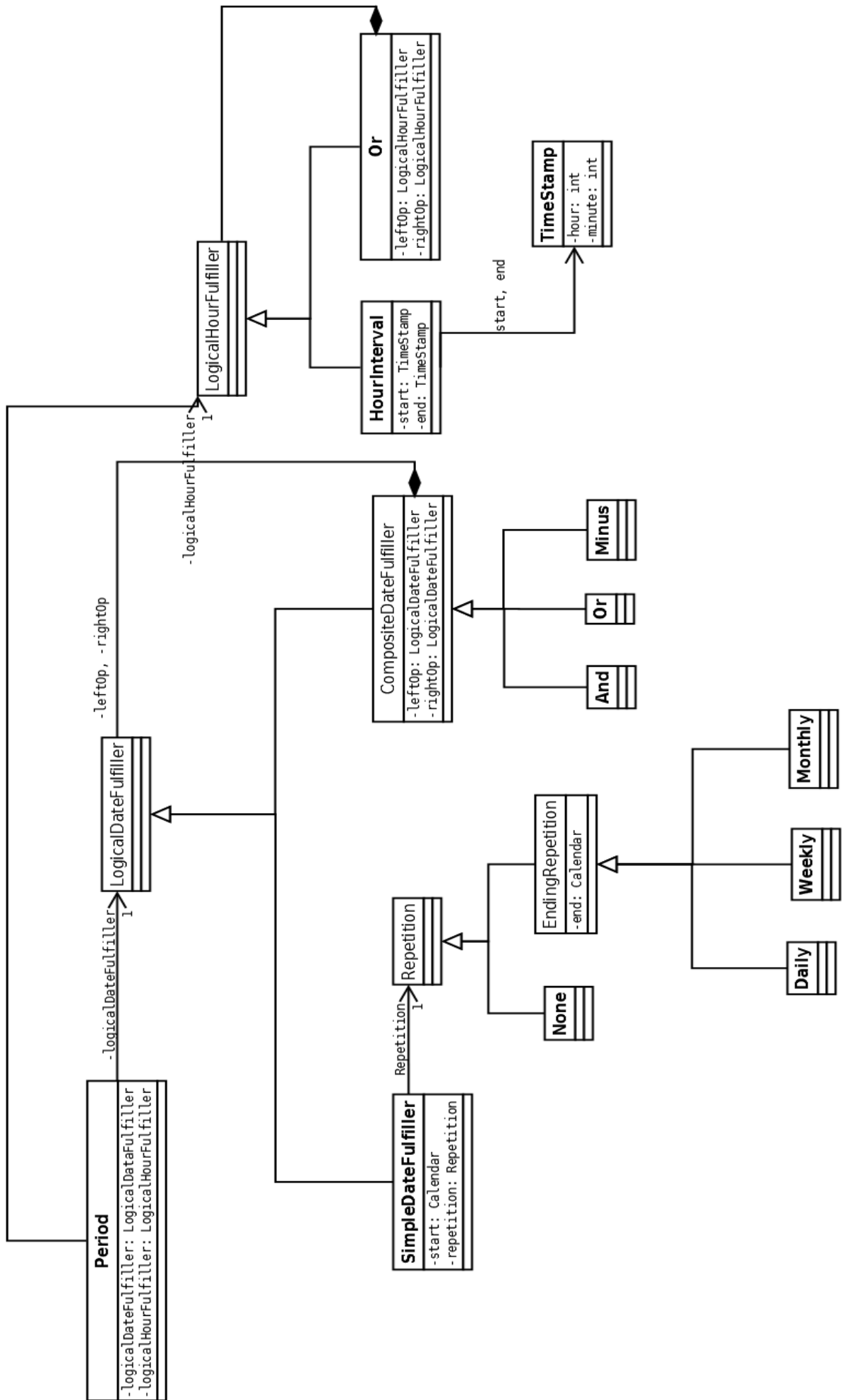
S.A.S.A. : Sistema de Asignación Sobre Aulas

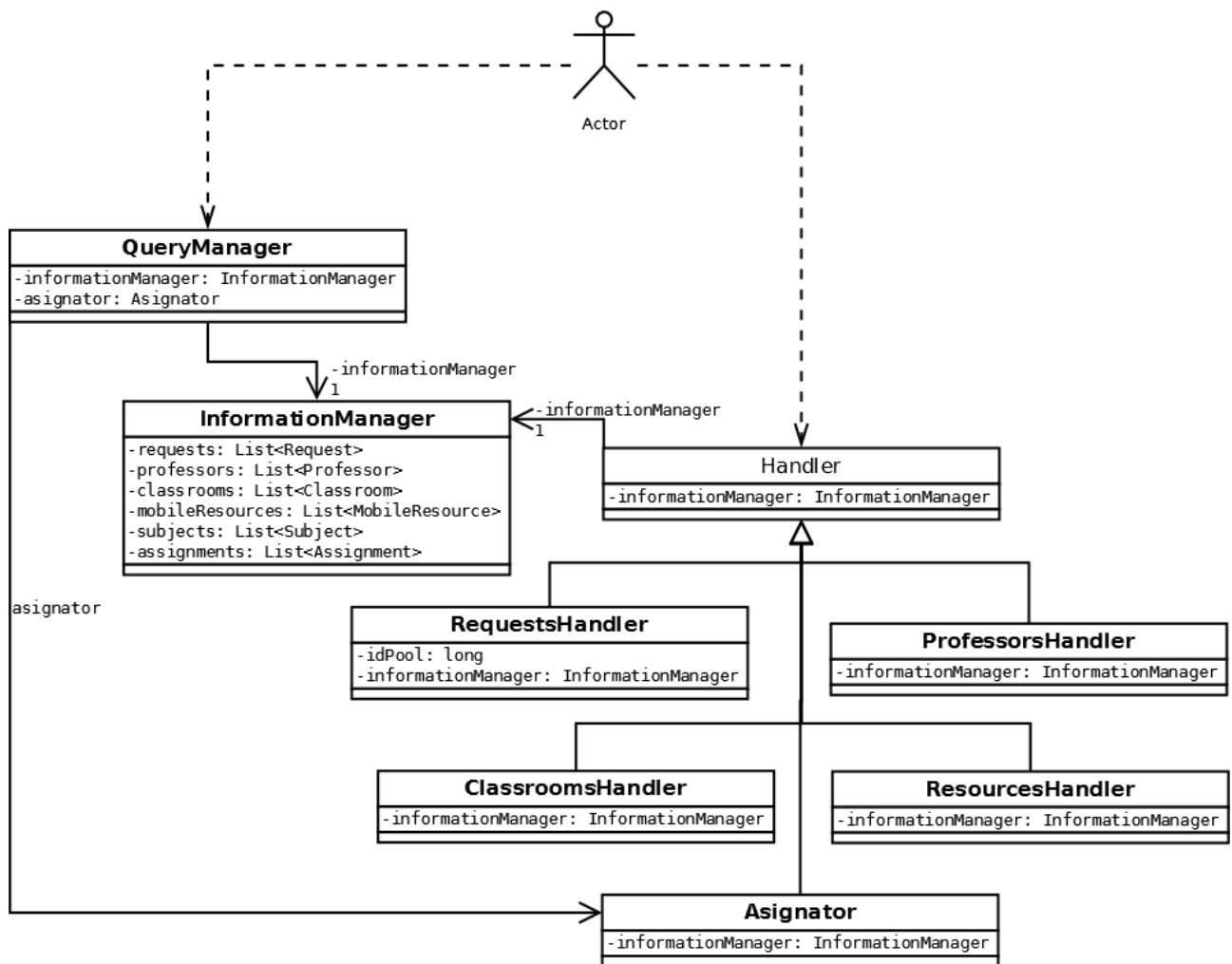
The main purpose of this system is to have a easy-to-use aplication for resources asignation and handling. The professors can perform classroom and resources requests, and receive the status of each one instantly. They also can make special requests -see the UML to note specifications-.

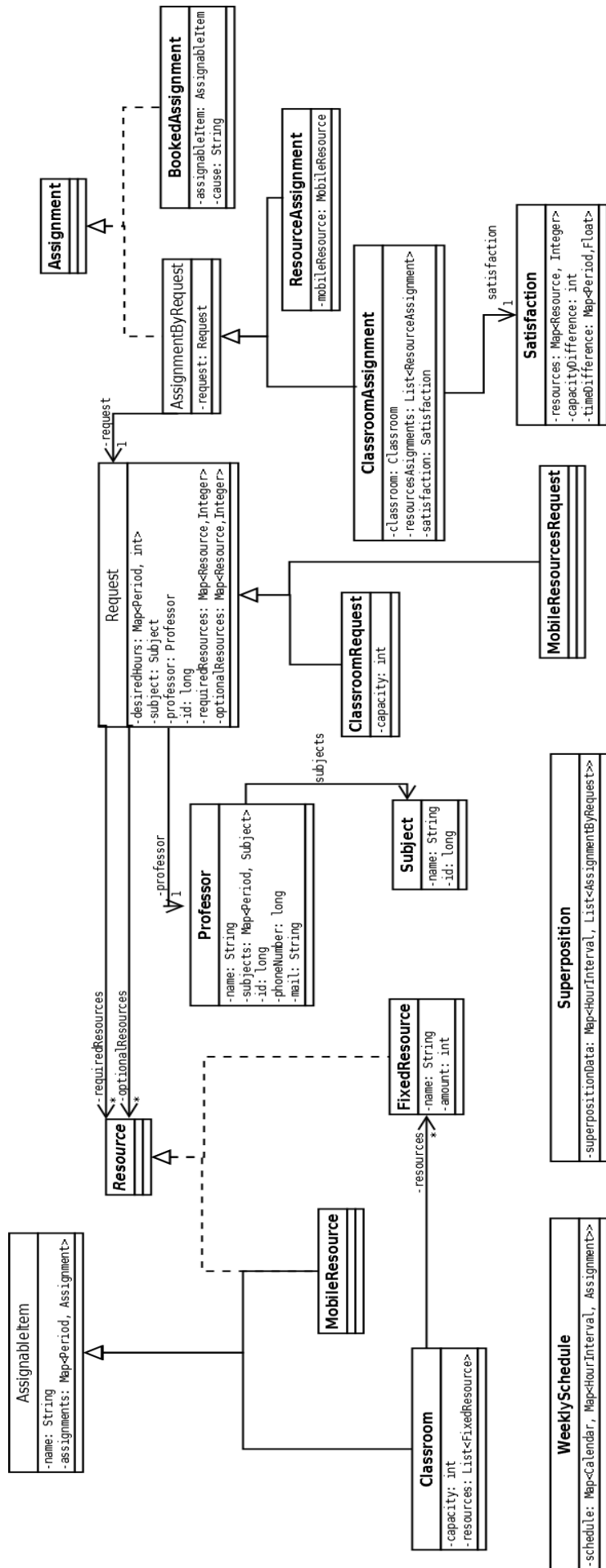
The actor is going to interact with classes that have his purpose focused on build and retrieve information about the resources already uploaded in the system.

The time management and implementation follows a logical structure.









Classes and methods

Handler

+Handler(InformationManager) : Handler
+getInformationManager() : InformationManager

RequestsHandler

+RequestsHandler(InformationManager) : RequestsHandler
+createClassroomRequest(Map<Resource,Integer> requiredResources, Map<Resource,Integer> optionalResources, Map<Period,Integer> desiredHours, Subject, Professor, int capacity) throws RequestException : ClassroomRequest
+createMobileResourcesRequest(Map<Resource,Integer> requiredResources, Map<Resource,Integer> optionalResources, Map<Period,Integer> desiredHours, Subject, Professor) : MobileResourcesRequest
+deleteRequest(Request) : void
+modifyRequest(Request, Map<Period, Integer> hours) : void
+modifyRequest(Request, Subject) : void
+modifyRequest(ClassroomRequest, int capacity) throws RequestException : void
+modifyRequestRequiredResources(Request, Map<Resource, Integer> requiredResources) : void
+modifyRequestOptionalResources(Request, Map<Resource, Integer> optionalResources) : void
+searchForRequest(Professor, Subject) : Request
+getInformationManager() : InformationManager

ResourcesHandler

+ResourcesHandler(InformationManager) : ResourcesHandler
+createMobileResource(String) : MobileResource
+createFixedResource(String, int) throws ResourceException : FixedResource
+deleteResource(String) : void
+modifyResource(Resource, String) : void
+modifyResource(FixedResource, int) throws ResourceException : void
+addMobileResourceAssignment(MobileResource, Period, ResourceAssignment) : void
+removeMobileResourceAssignment(MobileResource, Period) : void
+searchForResource(String) : Resource
+searchForPeriods(MobileResource) : Collection<Period>
+getInformationManager() : InformationManager

ClassroomsHandler

public ClassroomHandler(InformationManager)
+createClassroom(String, int) throws ClassroomException : void
+deleteClassroom(Classroom) : void
+deleteClassroom(String, int) throws ClassroomException : void
+searchClassroom(String, int) throws ClassroomException : Classroom

ProfessorsHandler

public ProfessorHandler(InformationManager aInformationManager)
+createProfessor(String, long, long, String) : void
+deleteProfessor(Professor) : void
+deleteProfessor(long) throws ProfessorException : void
+searchProfessor(long) throws ProfessorException : Professor

Asignator

+Asignator(InformationManager) : Asignator
+modifyBookedAssignmentsCause(BookedAssignment, String) : void
+searchForAssignment(AssignableItem, Period) : Assignment


```

+assignateBookedAssignment(AssignableItem, String, Period) : BookedAssignment
+assignateResourceAssignment(MobileResourcesRequest, MobileResource, Period) :
ResourceAssignment
+assignateClassroomAssignment(ClassroomRequest, Classroom, Period) :
ClassroomAssignment
+assignateRequestInMostSatisfactoryClassroom(ClassroomRequest) throws
PeriodException : void
+assignateRequestInAClassroom(ClassroomRequest, Classroom) throws PeriodException
: ClassroomAssignment
+deleteResourceAssignment(ResourceAssignment) : void
+deleteClassroomAssignmentFromARequest(Request) throws AssignmentException :
void
+deleteBookedAssignment(BookedAssignment, Period) : void
public void moveAssignmentOfClassroom(ClassroomAssignment, Classroom) throws
AssignmentException
public void moveAssignmentOfHour(ClassroomAssignment, LogicalHourFulfiller)
public void moveAssignmentOfDate(ClassroomAssignment, LogicalDateFulfiller)

```

InfomationManager

```

public InformationManager() : InformationManager
+getAssignments() : List<Assignment>
+getRequests() : List<Request>
+getProfessors() : List<Professor>
+getClassrooms() : List<Classroom>
+getMobileResources() : List<MobileResource>
+addRequest(Request) : void
+addProfessor(Professor) : void
+addClassroom(Classroom) : void
+addAssignment(Assignment) : void
+addResource(MobileResource) : void
+deleteAssignment(Assignment) : void
+deleteRequest(Request) : void
+deleteClassroom(Classroom) : void
+deleteResource(String) throws ResourceException : void
+getAssignmentByRequest(Request) throws AssignmentException : Assignment
+getClassroom(String) : Classroom
+getResource(String) : MobileResource
+getRequest(int) throws RequestException : Request
+getNResources(Resource, int) : List<MobileResource>

```

QueryManager

```

+QueryManager(InformationManager, Asignator) : QueryManager
+getInformationManager() : InformationManager
+getAsignator() : Asignator
+classroomsThatSatisfyTheWholeRequest(ClassroomRequest) throws PeriodException :
Collection<Classroom>
+classroomsThatSatisfyARequestRegardlessOfTheirAssignments(ClassroomRequest)
throws PeriodException : Collection<Classroom>
+classroomsThatSatisfyTimeRequirements(ClassroomRequest, boolean1) throws
PeriodException : Collection<Classroom>
+classroomsThatSatisfyCapacityRequirement(ClassroomRequest) :
Collection<Classroom>
+freeHoursInAnAssignnableItemInAGivenWeek(AssignableItem, Calendar) throws
PeriodException, TimestampException : Collection<Classroom>
+freeHoursInADay(AssignableItem, Calendar) throws TimestampException,
PeriodException : Period
+satisfactionsFromClassroomAndRequest(ClassroomRequest, Classroom) throws
PeriodException : Collection<Satisfaction>

```

¹ Boolean according to the truth value of "ignore common assignments"

```

+freeClassroomsInARequestGivenPeriod(Request) : Map<Classroom, Satisfaction>
+obtainRequestFromAnAssignment(AssignmentByRequest) : Request
+classroomsSuperpositions() : List<Superposition>
+weeklyScheduleOfAClassroom(Classroom) : WeeklySchedule
+weeklyScheduleOfAllClassrooms() : Map<Classroom, WeeklySchedule>

```

WeeklySchedule

```

+getSchedule()

```

Period

2

```

+Period(LogicalDateFulfiller, LogicalHourFulfiller) : Period
+getHourFulfiller() : LogicalHourFulfiller
+getDateFulfiller() : LogicalDateFulfiller
+setHourFulfiller(LogicalHourFulfiller) : void
+setDateFulfiller(LogicalDateFulfiller) : void
+addDateWithOrCondition(LogicalDateFulfiller) : void
+addDateWithAndCondition(LogicalDateFulfiller) : void
+subtractCondition(LogicalDateFulfiller) : void
+addHourCondition(LogicalHourFulfiller) : void
+intersectsWith(Period) : boolean
+contains(Calendar) : boolean throws PeriodException
+contains(Period) : boolean
+dividePeriodByRange(int minutes) : List<Period> throws PeriodException
+minutesSharedWith(Period) : int minutes3
+copy() : Period

```

CalendarUtils⁴

```

+compareEquals(Calendar, Calendar) : boolean5
+compareLess(Calendar, Calendar) : boolean
+compareGreater(Calendar, Calendar) : boolean

```

LogicalHourFulfiller

```

+contains(Timestamp) : boolean
+contains(LogicalHourFulfiller) : boolean
+intersectsWith(LogicalHourFulfiller) : boolean
+getHourIntervalsByRange(int minutes) : List<HourInterval>
+minutesSharedWith(LogicalHourFulfiller) : int

```

Or (for hours)

```

+Or(LogicalHourFulfiller, LogicalHourFulfiller) : Or
+contains(Timestamp) : boolean
+contains(LogicalHourFulfiller) : boolean
+intersectsWith(LogicalHourFulfiller) : boolean
+getHourIntervalsByRange(int minutes) : List<HourInterval>
+minutesSharedWith(LogicalHourFulfiller) : int

```

HourInterval

```

+HourInterval(Timestamp, Timestamp) : HourInterval
+getStart() : Timestamp
+getEnd() : Timestamp
+contains(Timestamp) : boolean
+contains(LogicalHourFulfiller) : boolean
+intersectsWith(LogicalHourFulfiller) : boolean
+getHourIntervalsByRange(int) : List<HourInterval>
+minutesSharedWith(LogicalHourFulfiller) : int

```

2 This class implements the following constant: MIN_HOUR_BLOCK = 30

3 This return value the superposed minutes between both periods.

4 Thi Class is not in the UML diagram; his purpose is to simplify the use and comparison between Calendars.

5 Static methods that only consider the day, month and year of the Calendars.

LogicalDateFulfiller

```
+intersectsWith(LogicalDateFulfiller) : boolean  
+contains(Calendar) : boolean  
+contains(LogicalDateFulfiller) : boolean  
public abstract LogicalDateFulfiller copy();
```

CompositeDateFulfiller

```
+CompositeDateFulfiller(LogicalDateFulfiller, LogicalDateFulfiller) :  
CompositeDateFulfiller  
+getLeftOp() : LogicalDateFulfiller  
+getRightOp() : LogicalDateFulfiller
```

Or

```
Or(LogicalDateFulfiller, LogicalDateFulfiller) : Or  
+intersectsWith(LogicalDateFulfiller) : boolean  
+contains(Calendar) : boolean  
+contains(LogicalDateFulfiller) : boolean  
+copy() : Or
```

And

```
public And(LogicalDateFulfiller, LogicalDateFulfiller) : And  
+boolean contains(LogicalDateFulfiller) : boolean  
+boolean contains(Calendar) : boolean  
+boolean intersectsWith(LogicalDateFulfiller) : boolean  
+copy() : And
```

Minus

```
+Minus(LogicalDateFulfiller, LogicalDateFulfiller) : Minus  
+intersectsWith(LogicalDateFulfiller) : boolean  
+contains(Calendar) : boolean  
+contains(LogicalDateFulfiller) : boolean  
+copy() : Minus
```

SimpleDateFulfiller

```
+SimpleDateFulfiller(Calendar) : SimpleDateFulfiller6  
+SimpleDateFulfiller(Calendar, Repetition) : SimpleDateFulfiller  
+getStartDate() : Calendar  
+getRepetition() : Repetition  
+intersectsWith(LogicalDateFulfiller) : boolean  
+contains(Calendar) : boolean  
+contains(LogicalDateFulfiller) : boolean  
+copy() : SimpleDateFulfiller
```

Repetition

```
+containsInSomeRepetition(Calendar, Calendar) : boolean  
+thereIsSomeDayIn(SimpleDateFulfiller, Calendar) : boolean  
+isAllDaysIn(SimpleDateFulfiller, Calendar) : boolean
```

EndingRepetition

```
+EndingRepetition(Calendar) : EndingRepetition  
+getEndDate() : Calendar  
+containsInSomeRepetition(Calendar, Calendar) : boolean  
+thereIsSomeDayIn(SimpleDateFulfiller, Calendar) : boolean  
+isAllDaysIn(SimpleDateFulfiller, Calendar) : boolean  
Timestamp  
+Timestamp(int, int) : Timestamp  
+Timestamp(int) : Timestamp7  
+getHour() : int
```

⁶ This constructor only takes the initial date, and sets the repetition as a None repetition.

⁷ This constructor only takes the hours, and sets the minutes as 0.

```

+getMinutes() : int
+setHour(int) : void
+setMinutes(int) : void
+equals(Timestamp) : boolean8
+greaterThan(Timestamp) : boolean
+lessThan(Timestamp) : boolean
+greaterEqual(Timestamp) : boolean
+lessEqual(Timestamp) : boolean
+add(int minutes) : Timestamp9
+subtract(int minutes) : Timestamp10
+minutesBetween(Timestamp) : int11

```

None

```

+containsInSomeRepetition(Calendar, Calendar) : boolean
+isAllDaysIn(SimpleDateFulfiller, Calendar) : boolean
+thereIsSomeDayIn(SimpleDateFulfiller, Calendar) : boolean

```

Daily

```

+Daily(Calendar) : Daily

```

Monthly

```

+Monthly(Calendar) : Monthly

```

Weekly

```

+Weekly(Calendar) : Weekly

```

Request

```

+getRequiredResources() : Map<Resource,Integer>
+getOptionalResources() : Map<Resource,Integer>
+getProfessor() : Professor
+getSubject() : Subject
+getDesiredHours() : Map<Period,int>

```

ClassroomRequest

```

+ClassroomRequest(Map<Period,Integer>,Subject,Professor, int) throws
RequestException : ClassroomRequest
+getCapacity() : int

```

MobileResourcesRequest

```

+MobileResourcesRequest(Map<Period,Integer>,Subject,Professor, int,
Map<Resource,Integer>,Map<Resource,Integer>) : MobileResourcesRequest

```

Professor

```

+Professor(String, long,long, String): Professor
+getName() : String
+setName(String) : void
+getSubjects() : List<Subject>
+getID() : long
+getPhoneNumber() : long
+setPhoneNumber(long) : void
+getMail() : String
+setMail(String) : void
+addNewSubject(Subject): void
Subject
+getID(): long
+getName(): String
+setName(String): void

```

8 Two Timestamp are equals if his hours and minutes are it.

9 A new instance is created, it doesn't modifies the target object.

10 A new instance is created, it doesn't modifies the target object.

11 Returns the number of minutes between the receiver and the parameter.

SuperPosition

```
+Superposition() : Superposition  
+getSuperpositionData() : Map<HourInterval, List<AssignmentByRequest>>
```

Resource *interface*

```
String getName();  
void setName(String);  
boolean isFixedResource();
```

FixedResource

```
+FixedResource(String, int) : FixedResource  
+getAmount() : int  
+setAmount(int) : void  
+getName() : String  
+setName(String) : void  
+isFixedResource() : boolean
```

MobileResource

```
+MobileResource(String) : MobileResource  
+setName(String) : void  
+isFixedResource() : boolean
```

AssignableItem

```
+AssignableItem(String) : AssignableItem  
+getName() : String  
+setName(String) : void  
+getAssignments() : Map<Period, Assignment>  
+addAssignment(Period, Assignment) : void  
+removeAssignment(Period) : void  
+satisfyTimeRequirements(Request, boolean12) throws PeriodException : boolean  
+satisfyTimeRequirements(Request) throws PeriodException : boolean  
+canAssign(Period period, boolean13) : boolean  
+canAssign(Period period) : boolean  
+isFreeAt(Calendar) throws PeriodException : boolean
```

Classroom

```
+Classroom(String, int) : Classroom  
+getResources() : List<FixedResources>  
+addResource(FixedResource) : void  
+getCapacity() : int  
+setCapacity(int) : void  
+hasResource(String) : boolean  
+getResource(String) throws ResourceException : FixedResource  
+getAssignment(Period) : ClassroomAssignment  
+satisfyFixedResources(Map<FixedResource, Integer>) : boolean  
+satisfyFixedResource(FixedResource, int) : boolean
```

Assignment *interface*

```
+getAssignableItem() : AssignableItem  
+getRequest() : Request
```

AssignmentByRequest

```
+AssignmentByRequest(Request) : AssignmentByRequest  
+getRequest() : Request
```

BookedAssignment

```
+BookedAssignment(String, AssignableItem) : BookedAssignment  
+getCause() : String  
+setCause(String) : void
```

¹² Boolean according to the truth value of "ignore common assignments".

¹³ Boolean according to the truth value of "ignore common assignments".

```
+getRequest() : null  
+getAssignableItem() : AssignableItem
```

ClassroomAssignment

```
+ClassroomAssignment(ClassroomRequest, Classroom, Collection<ResourceAssignment>)  
: ClassroomAssignment  
+createSatisfaction() : Satisfaction  
+getSatisfaction() : Satisfaction  
+getAssignableItem() : Classroom  
+setAssignableItem(Classroom) : void  
+getRequest() : ClassroomRequest
```

ResourceAssignment

```
+ResourceAssignment(MobileResource) : ResourceAssignment  
+getAssignableItem() : MobileResource  
+getRequest() : Request
```

Satisfaction

```
+Satisfaction(Map<Resource, Boolean>, Map<Resource, Integer>, HourInterval) :  
Satisfaction  
+getSingleResources() : Map<Resource, Boolean>  
+getMultipleResources() : Map<Resource, Integer>  
+getTimeDifference() : HourInterval
```

Workgroup divisions

Division 1 – by Diego Campos

Classes: *ClassroomsHandler, ProfessorsHandler, Handler, ResourceHandler, AssignableItems, Classroom, MobileResource, Resource, FixedResource, TestClassroom.*

Methods: *moveAssignmentOfClassroom(ClassroomAssignment, Classroom)*,
,moveAssignmentOfHour(ClassroomAssignment, LogicalHourFulfiller),
moveAssignmentOfDate(ClassroomAssignment , LogicalDateFulfiller) from the Assignator class. He also contributed in the Assignator testing.

Division 2 – by Gastón Charkiewicz

Classes: *WeeklySchedule, Superposition, Subject, Professor, Request, ClassroomRequest, MobileResourcesRequest, TestMobileResourcesRequest, TestWeeklySchedule, TestSuperposition, TestSubject, TestProfessor, TestClassroomRequest.*

Methods: *classroomsSuperpositions(), weeklyScheduleOfAClassroom(Classroom), weeklyScheduleOfAllClassrooms()* from the QueryManager class.

Others: presentation document.

Division 3 – by Nahuel Garbezza

Classes: *Period, SimpleDateFulfiller, Or, And, Minus, Repetition, None, Daily, Weekly, Monthly, HourInterval, Or, Timestamp, TestPeriod, TestSimpleDateFulfiller, TestNone, TestDaily, TestWeekly, TestMonthly, TestAnd, TestMinus, TestOr, TestTimestamp, TestHourInterval, TestOr.*

Methods: *classroomsThatSatisfyTheWholeRequest(ClassroomRequest), classroomsThatSatisfyARequestRegardlessOfTheirAssignments(ClassroomRequest), freeHoursInAnAssignableItemInAGivenWeek(AssignableItem, Calendar)* from the QueryManager class. He also contributed in the QueryManager testing.

Others: Javadoc.

Division 4 – by Cristian Suárez

Classes: *Assignment, AssignmentByRequest, BookedAssignment, ClassroomAssignment, ResourceAssignment, Satisfaction, Handler, RequestsHandler, ResourcesHandler, TestBookedAssignment, TestClassroomAssignment, TestResourceAssignment, TestSatisfaction, TestRequestsHandler, TestResourcesHandler.*

Methods: *satisfactionsFromClassroomAndRequest(ClassroomRequest, Classroom)* from the QueryManager class; *modifyBookedAssignmentsCause(BookedAssignment, String), searchForAssignment(AssignableItem, Period), assignateBookedAssignment(AssignableItem, String, Period), assignateResourceAssignment(MobileResourcesRequest, MobileResource, Period), assignateClassroomAssignment(ClassroomRequest, Classroom, Period), assignateRequestInMostSatisfactoryClassroom(ClassroomRequest), assignateRequestInAClassroom(ClassroomRequest, Classroom), deleteResourceAssignment(ResourceAssignment), deleteClassroomAssignmentFromARequest(Request request), deleteBookedAssignment(BookedAssignment, Period), searchForAssignment(AssignableItem, Period)* from the Assignator class. He also contributed in the Assignator testing.