DETI | ua

universidade de aveiro

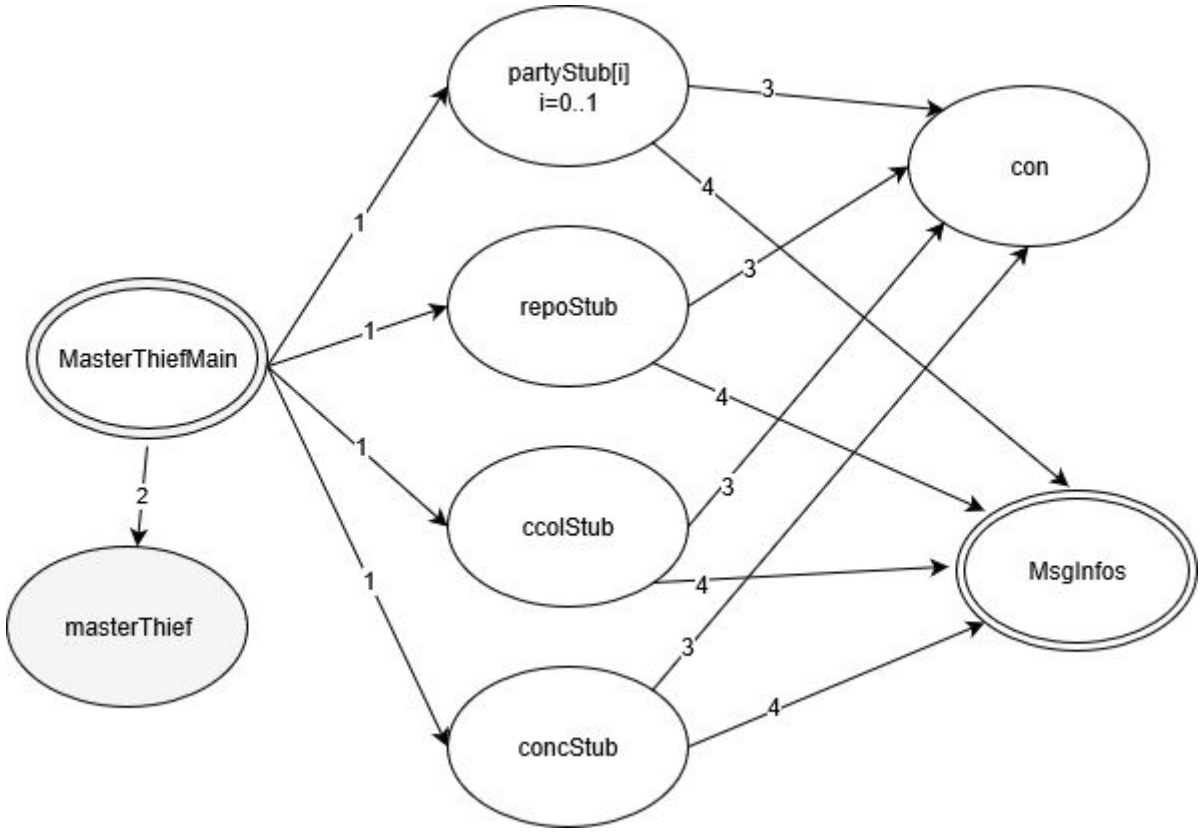
Departamento Electrónica, Telecomunicações e Informática

Computação Distribuída
Trabalho Prático Nº2

Trabalho realizado por:
André Cardoso, 65069 (marquescardoso@ua.pt)
Dércio Bucuane, 83457 (dbucuane@ua.pt)

Diagramas do lado do Cliente:

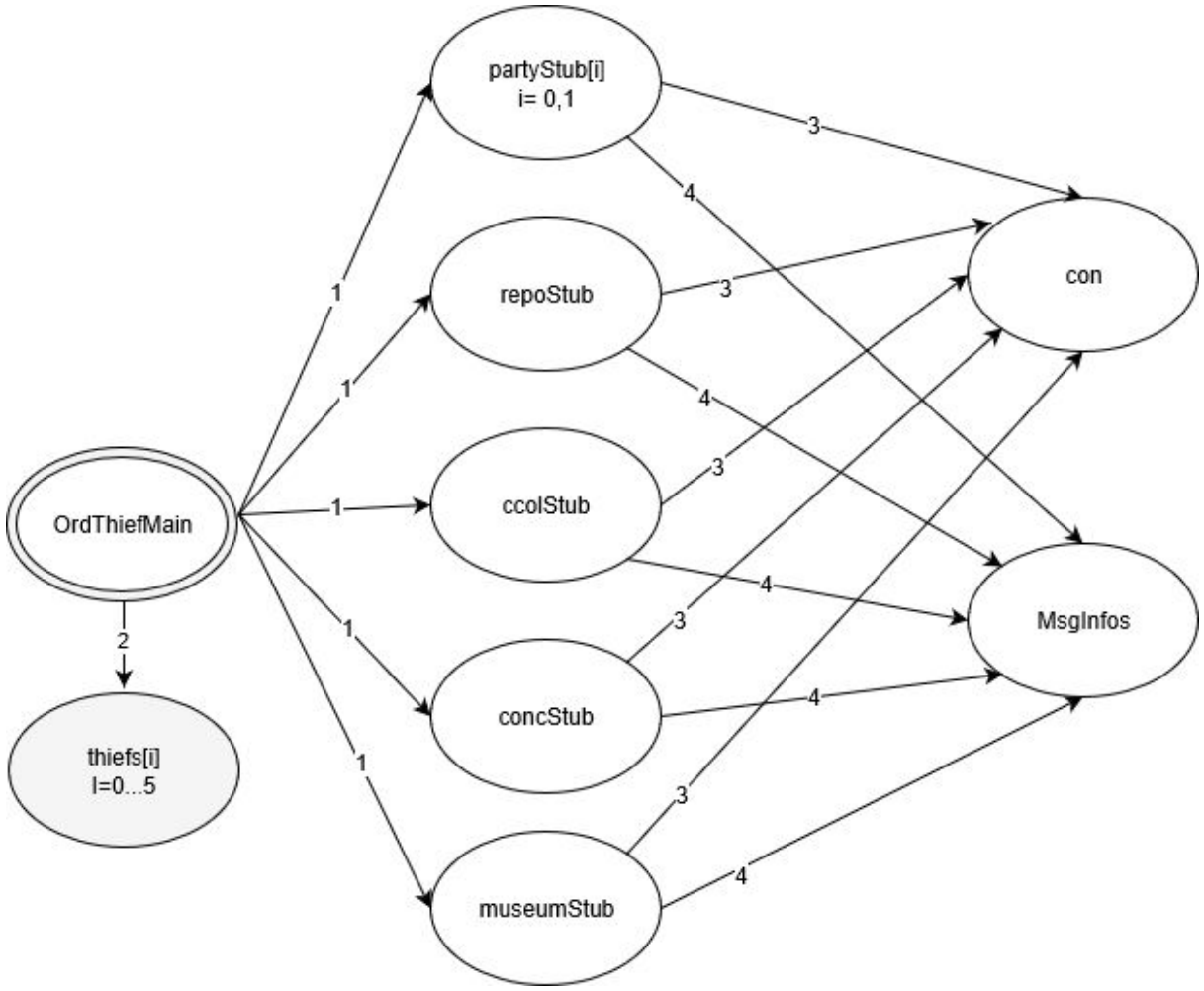
Master Thief



Legenda:

1 - Instância;
2 - Instância e lança Thread;
3 - writeObject, readObject;
4 - Ips, Ports, Message;

Ordinary Thief

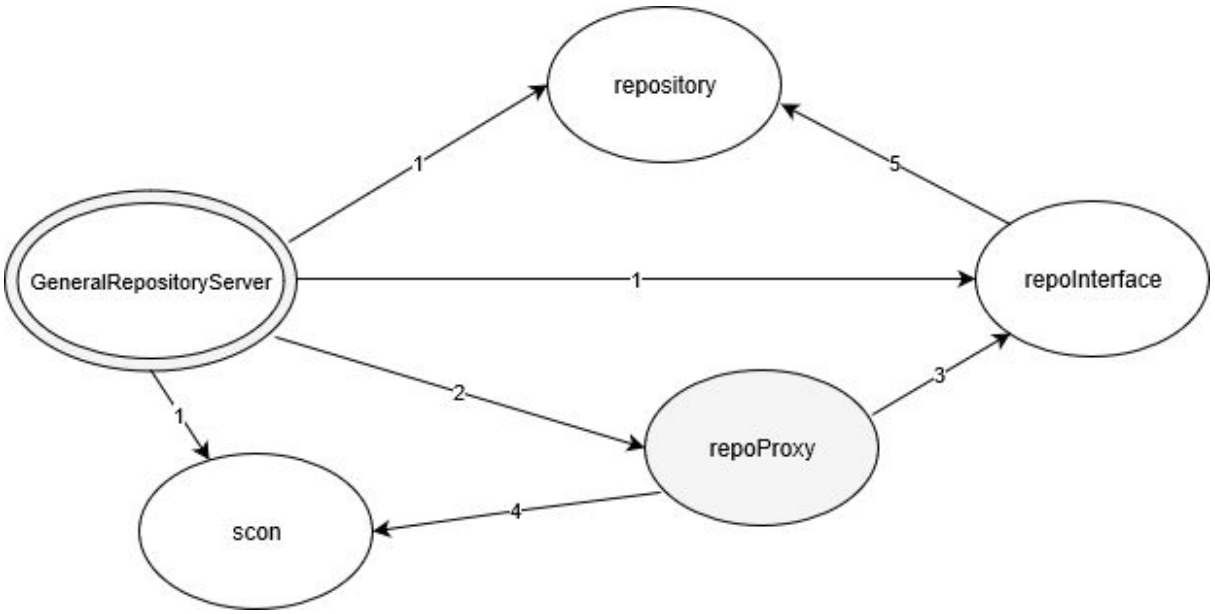


Legenda:

1 - Instância;
2 - Instância e lança Thread;
3 - writeObject, readObject;
4 - Ips, Ports, Message;

Diagramas do lado do Servidor:

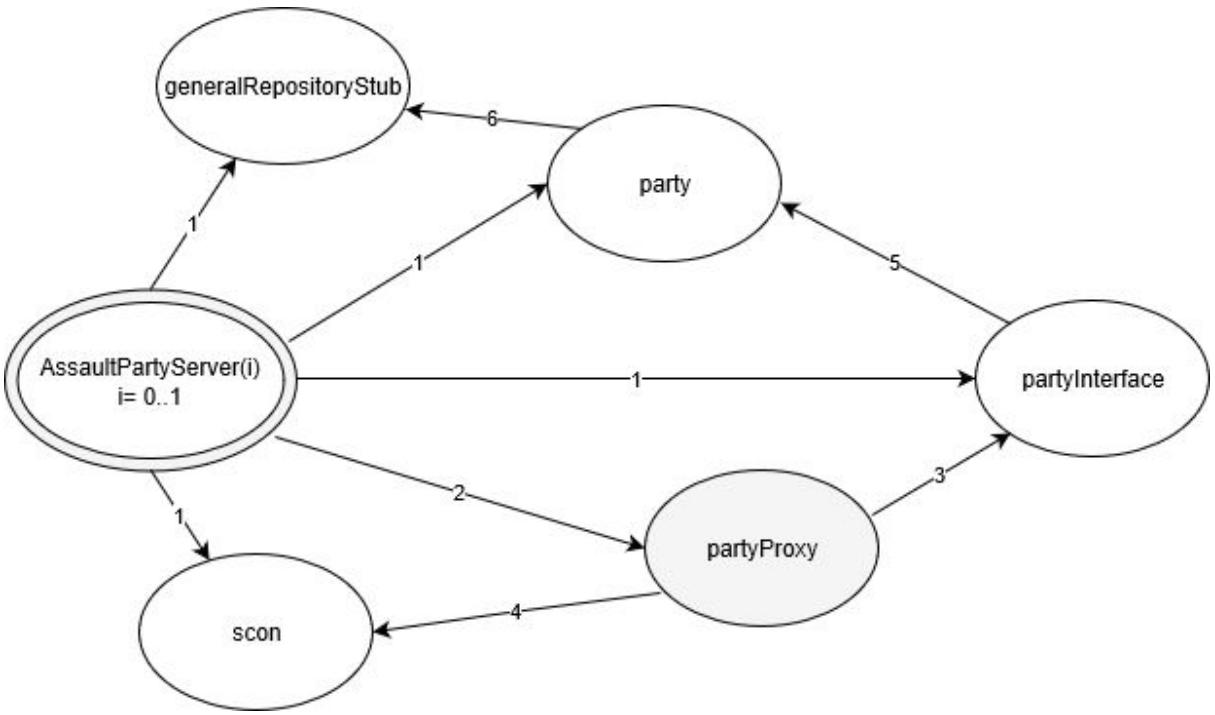
General Repository Server:



Legenda:

1 - Instância;
2 - Instância e lança Thread
3 - processAndReply;
4 - readObject, writeObject;
5 - reportStatus

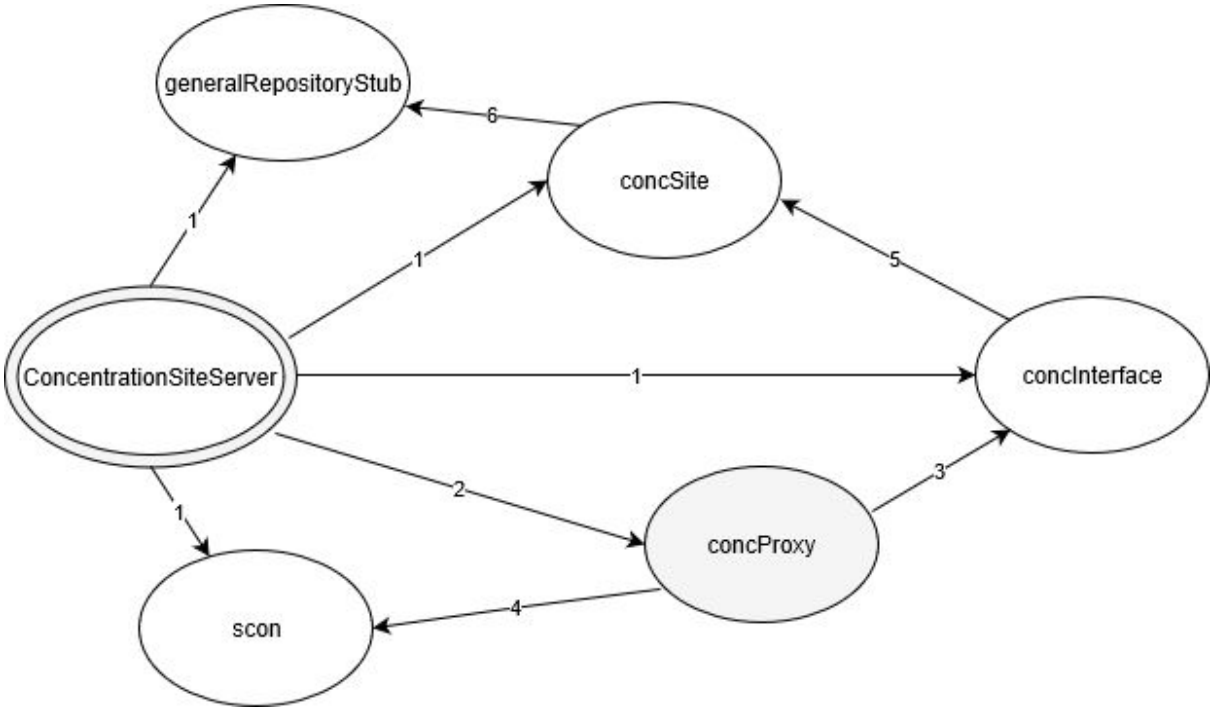
Assault Party Server:



Legenda:

1 - Instância;
2 - Instância e lança Thread
3 - processAndReply;
4 - readObject, writeObject;
5 - prapareExcursion, crawlin, crawlout, reverseDirection, sendAssaultParty, getRoomId;
6 - reportStatus

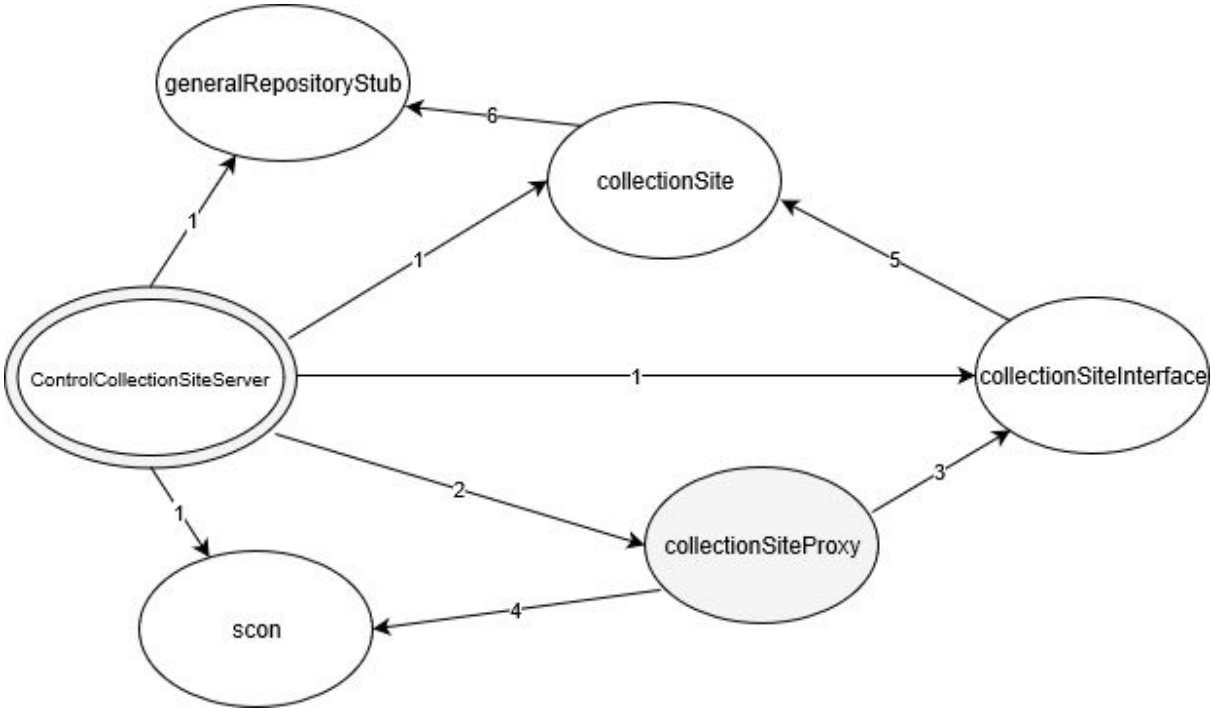
Concentration Site Server:



Legenda:

1 - Instância;
2 - Instância e lança Thread
3 - processAndReply;
4 - readObject, writeObject;
5 - amINeeded, prepareAssaultParty, sumUpResults;
6 - reportStatus

Control Collection Site Server:



Legenda:

1 - Instância;

2 - Instância e lança Thread

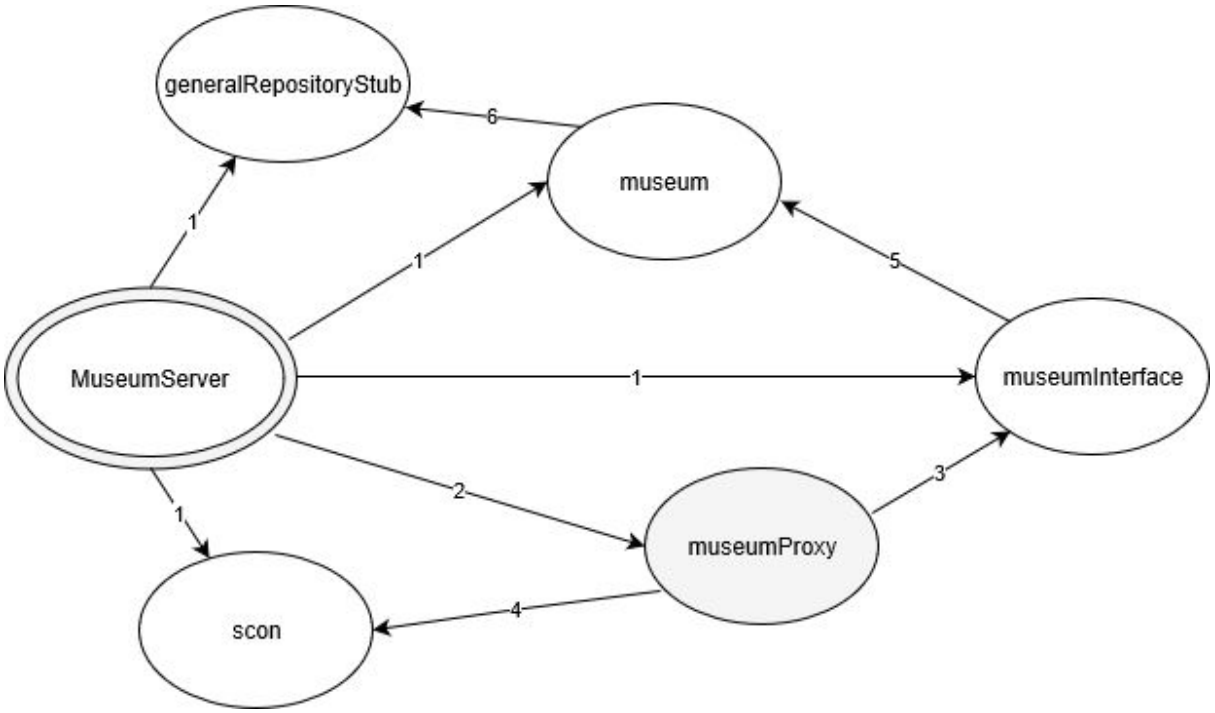
3 - processAndReply;

4 - readObject, writeObject;

5 - startOperations, appraiseSit, prepareAssaultParty_1, prepareAssaultParty_2, takeRest, collectCanvas, handACanvas, sumUpResults;

6 - reportStatus

Museum Server:



Legenda:

1 - Instância;
2 - Instância e lança Thread
3 - processAndReply;
4 - readObject, writeObject;
5 - rollACanvas, getRoomDistance, getNumCanvas;
6 - reportStatus

Scripts

O script principal é TheHeistDeployAndRun.sh, que por sua vez este executa outros scripts encontrados no mesmo diretório em que cada um deles, como o nome indica, lida com um servidor da região partilhada ou cliente da threads identidades. Depois de iniciados os servers, os serviços de todas as regiões partilhadas ficam estabelecidos e fiquem à escuta. De seguida são lançados as entidades (Master Thief e os Ordinary Thieves).

Todos os scripts estão preparados para apagar no servidor caso exista o ficheiro .zip antigo, inserir o código fonte ficheiro .zip e enviar para o servidor,. E por último é compilado e executado ficheiros gerados.

Tipo	Nome	Maquina	script
Região Partilhada	General Repository	ws01	GeneralRepositoryDeployAndRun.sh
	Assault Party 1	ws09	AssaultParty1DeployAndRun.sh
	Assault Party 2	ws03	AssaultParty2DeployAndRun.sh
	Concentration Site	ws04	ConcentrationSiteDeployAndRun.sh
	Control Collection Site	ws05	ControlCollectionSiteDeployAndRun.sh
	Museum	ws07	MuseumDeployAndRun.sh
Entidade	Master Thief	ws10	MasterThiefDeployAndRun.sh
	Ordinary Thieves	ws10	OrdThiefDeployAndRun.sh

Nota:

ws02, ws06 e ws08 não estavam em funcionamento.

Descrição da transformação em mensagens os métodos invocados nas regiões partilhadas.

General Repository

Método	Mensagem de Envio	Mensagem de Resposta
void setState(MasterThiefStates)	Id = SM_SETSTATEMT state = MasterThiefState	Id = SM_SETSTATEMT
void setState(OrdThiefStates)	Id = SM_SETSTATEMT state = OrdThiefState	Id = RM_SETSTATET
void setSituation (OrdThiefSituation)	Id = SM_SETSITUATION State = OrdThiefSituation	Id =
void initializeRepository()	Id = SM_INITIALIZEREPOSITORY	Id = RM_INITIALIZEREPOSITORY
void setRollACanvas(int)	Id = SM_SETROLLACANVAS inParam = roomId	Id = RM_SETROLLACANVAS
void initializePartyProperties(int)	Id = SM_INITIALIZEPARTYPROPERTIES inParam = assaultPartyId	Id = RM_INITIALIZEPARTYPROPERTIES
void setCarryingCanvas(int, int)	Id = SM_SETCARRYINGCANVAS inParam = carryingCanvas nParam = carryingCanvas	Id = RM_SETCARRYINGCANVAS
int getCarryingCanvas(int)	Id = SM_getCarryingCanvas idSender = idThief	Id = RM_getCarryingCanvas inParam = carry canvas situation]
void setPosition(int, integer)	Id = SM_setPosition idSender = idThief inParam = position	Rm = RM_setPosition
void setAssaultPartyRoom(int, int)	Id = SM_setAssaultPartyRoom idSender = idThief inParam = roomId	Id = RM_setAssaultPartyRoom
int getNumWaitingThieves()	Id = SM_getNumWaitingThieves	Id = RM_getNumWaitingThieves

int getAssaultParty1Members()	Id = SM_GETASSAULTPARTY1MEMBERS	Id = RM_GETASSAULTPARTY1MEMBERS
int getAssaultParty2Members()	Id = SM_GETASSAULTPARTY2MEMBERS	Id = RM_GETASSAULTPARTY2MEMBERS
void setAssaultPartyMember(int)	Id = SM_SETASSAULTPARTYMEMBER	Id = RM_SETASSAULTPARTYMEMBER
void removeAssaultPartyMember (int)	Id = SM_REMOVEASSAULTPARTYMEMBER	Id = RM_REMOVEASSAULTPARTYMEMBER
void reportStatus()	Id = SM_REPORTSTATUS	Id = RM_REPORTSTATUS
void reportFinalStatus (int)	Id = SM_reportFinalStatus	Id = RM_reportFinalStatus
void reportInitialStatus()	Id = SM_reportInitialStatus	Id = RM_reportInitialStatus

Assault Party

Método	**Mensagem de Envio**	**Mensagem de Resposta**
void prapareExcursion (int idThief)	Id = SM_PRAPAREEXCURSION inParam = idThief	Id = RM_PRAPAREEXCURSION
void crawlin(int idThief)	Id = SM_CRAWLIN inParam = idThief	Id = RM_CRAWLIN
void reverseDirection(int idThief)	Id = SM_REVERSEDIRECTIONN inParam = idThief	Id = RM_REVERSEDIRECTION
void crawlout(int idThief)	Id = SM_CRAWLOUT inParam = idThief	Id = RM_CRAWLOUT
void sendAssaultParty(int roomID)	Id = SM_SENDASSAULTPARTY inParam = roomID	Id = RM_SENDASSAULTPARTY
Integer getRoomId()	Id = SM_GETROOMID	Id = RM_GETROOMID inParam = roomId
void shutDown()	Id = SM_SHUTDOWN	Id = RM_SHUTDOWN

Concentration Site

Método	Mensagem de Envio	Mensagem de Resposta
Integer amINeeded (int idThief)	Id = SM_AMINEEDED idSender = isThief	Id = RM_AMINEEDED
void prepareAssaultParty (Integer assaultPartyID)	Id = SM_PREPAREASSAULTPARTY inParam = assaultPartyID	Id = RM_PREPAREASSAULTPARTY
void sumUpResults()	Id = SM_SUMUPRESULTS	Id = RM_SUMUPRESULTS

Control Collection Site

Método	Mensagem de Envio	Mensagem de Resposta
Integer amINeeded(int id)	Id = SM_AMINEEDED inParam = idThief	Id = RM_AMINEEDED inParam = roomId
void prepareAssaultParty(Integer assaultPartyID)	Id = SM_PREPAREASSAULTPARTY inParam = assaultPartyID	Id = RM_PREPAREASSAULTPARTY
void sumUpResults()	Id = SM_SUMUPRESULTS	Id = RM_SUMUPRESULTS

Museum

Método	Mensagem de Envio	Mensagem de Resposta
Integer rollACanvas (Integer *roomId*)	Id = SM_ROLLACANVAS inParam = *roomId*	Id = RM_ROLLACANVAS inParam = [0 se não existir quadros, 1 caso contrario]
int getRoomDistance (Integer *roomId*)	Id = SM_GETROOMDISTANCE inParam = *roomId*	Id = RM_GETROOMDISTANCE inParam = [distância do quarto]
int getNumCanvas (Integer	Id = SM_GETNUMCANVAS	Id = RM_GETNUMCANVAS

roomId)	inParam = *roomId*	inParam = [número de quadro no museu]
void shutDown()	Id = SM_SHUTDOWN	Id = RM_SHUTDOWN