Capsules
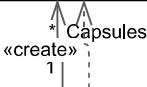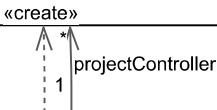
+ Capsules(int, String, String, String, String, String, String):

---

description: String
type: String
situation: String
status: String
stage: int
collaboratorName: String
identifier: String

*Capsules
«create»
1

Projects

+ Projects(String, String, String, String, Double, String[], String[],

---

+ aproveCapsule(String, String): void
+ getClientManagerPhones(): void
+ culminateStage(String): void
+ getClientManagerNames(): void
+ regCapsule(int, String, int, String, String, String, String, String,
+ publishCapsule(String): void
+ getCompanyManagerNames(): void
+ getCompanyManagerPhones(): void

---

name: String
realDateEnd: String
datePlannedBegin: String
client: String
datePlanedEnd: String
stage: int
budget: Double
stageActivation: int
realDateBegin: String

«create»
* projectController
1

Green

+ Green(int, int, String, String, String, String, double, int):

+ stageCulmination(String, String, int): void
+ aproveCapsule(String, String, String, int): void
+ publishCapsule(String, String, int): void
+ registerCapsule(String, int, String, String, String, String, String
+ stageValidation(String, int, int): void

«create»
greenController
1

GreenSQASystem

+ GreenSQASystem():

+ capsulePublishing(): void
+ main(String[]): void

Utils

+ Utils():

+ validateDoubleInput(): double
+ print(Object): void
+ validateHashtag(): String
+ validateIntInput(): int
+ validateStageInput(): int

+ stageCulmination(): void

+ menu(): void

+ projectCreation(): void

+ registerCapsule(): void