|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Nombre de la Clase** | **Nombre del método** |
| RF0: Creación de proyecto | Main | registerProyect |
| GreenSQA | * isRegistrableProyect() * registerProject(String projectName, String clientName, Calendar initialDate, Calendar finalDate, double budget, String proyectsManagerName, String proyectsManagerCellphone, String clientsManagerName, String clientsManagerCellphone) * getFirstValidPosition() * validateDouble(String message) * readDate(String msg) |
| Project | * Project(String name, String clientName, Calendar initialDate, Calendar finalDate, double budget, Manager proyectsManager, Manager clientsManager) |
| Manager | * Manager(String name, String cellphone) |
| RF1: Creación de las etapas del proyecto. | Main | * addPhase(int ProjectsPosition) * validateInteger(String message) |
| GreenSQA | * addPhases(int[] monthsPerPhase, int projectPosition) * setPhasesPlannedDates(int projectPosition, Calendar startDate, int[] monthsPerPhase) |
| Project | * addPhase(ProjectsPhase phase) * getPhase(int pos) |
| ProjectsPhase | * ProjectsPhase(int months, int phaseNameOption) * setStatus(boolean status) * setStartDate(Calendar date) * setPlannedStartDate(Calendar date) * setPlannedEndDate(Calendar date) |
| RF2: Culminación de una etapa del proyecto | Main | * culminatePhase() |
| GreenSQA | * searchProject(String projectSearch) * culminatePhase(int positionProject) |
| Project | * getName() * culminateActivePhase() |
| ProjectsPhase | * getStatus() * setStatus(boolean status) * setApprovalStatus(boolean isApproved) * setEndDate(Calendar date) * getNamePhase() * setStartDate(Calendar date) |
| RF3: Registrar cápsulas de conocimiento | Main | * isRegistrableCapsule * registerCapsule(int pos) * capsulesTypeMenu() * validateHashtags(String input) * validateInteger * capsulesTypeMenu() |
| GreenSQA | * searchProject(String projectSearch) * isProejctAvailable(int pos) * isPhaseCapsulesLimitReached(int pos) * isIdAlreadyRegistered(int pos, String idInput) * registerCapsule(int projecPosition, String id, String description, int capsuleTypeOption, String collaboratorsName, String collaboratorsPosition, String learnedLesson) |
| Project | * getActivePhase() * getPhase(int pos) * sendCapsuleToPhase (String id, String description, int capsuleTypeOption, String collaboratorsName, String collaboratorsPosition, String learnedLesson) |
| ProjectsPhase | * isCapsulesLimitReached() * getMaxCapsules() * getCapsule(int position) * addKnowledgeCapsule(KnowledgeCapsule capsule) |
| KnowledgeCapsule | * getId() * KnowledgeCapsule(String id, String description, Collaborator collaborator, String learnedLesson, int capsuleTypeOption) |
| Collaborator | * Collaborator(String name, String position) |
| RF4: Aprobación de las cápsulas | Main | * approveCapsule() |
| GreenSQA | * searchProject(String projectSearch) * approveCapsule(int projecPosition, String capsuleIdSearch) |
| Project | * getPhase(int pos) * getActivePhase() * getName() |
| ProjectsPhase | * searchKnowledgeCapsule(String searchCapsule) * getCapsule(int position) |
| KnowledgeCapsule | * getId() * setApprovalStatus(boolean newStatus) * setApprovalDate(Calendar date) |
| RF5: Publicación de las cápsulas a la organización. | Main | * publishCapsule() |
| GreenSQA | * searchProject(String projectSearch) * isCapsuleApprove (int projecPosition, String capsuleIdSearch) |
| Project | * getName() * getPhase(int pos) * getActivePhase() * getNamePhase() |
| ProjectsPhase() | * searchKnowledgeCapsule(String searchCapsule) * getCapsule(int position) |
| KnowledgeCapsule | * getApprovalStatus * getId() |
| RF6: Informar al usuario cuantas de las cápsulas registradas hay por cada tipo de cápsula | Main | * showAmountCapsulesPerType() |
| GreenSQA | * calculateProjectsCapsulePerType() |
| Project | * getMaxPhases() * getPhase(int pos) |
| ProjectsPhase | * getMaxCapsules() * getCapsule(int pos) |
| KnowledgeCapsule | * getType() |
| RF7: Informar al usuario un listado de lecciones aprendidas correspondientes a las cápsulas registradas en los proyectos para una etapa en particular | Main | * showPhaseLearnedLessons() * phaseTypeMenu() * validateInteger() |
| GreenSQA | * obtainPhasesLearnedLessons(int phasePosition) |
| Project | * getName() * getPhase(int pos) |
| ProjectPhase | * getMaxCapsules() * getCapsule() |
| KnowledgeCapsule | * getLearnedLesson() |
| RF8: Informar al usuario el nombre del proyecto con más cápsulas registradas | Main | * showProjectWithMostCapsules() |
| GreenSQA | * obtainProjectsNameWithMostCapsules() |
| Project | * getMaxPhases() * getPhase(int pos) * getName() |
| ProjectPhase | * getCapsulesCounter() |
| RF9: Informar al usuario si un colaborador ha registrado cápsulas en algún proyecto. | Main | * showCollaboratorsCapsulesa() |
| GreenSQA | * collaboratorHasRegisteredCapsule(String collaborators Name) |
| Project | * getMaxPhases() * getPhase(int pos) * getName() |
| ProjectPhase | * getMaxCapsules() * getCapsule(int position) |
| KnowledgeCapsule | * getCollaborator() * getId() |
| Collaborator | * getName() |
| RF10: Informar al usuario las situaciones y lecciones aprendidas de las cápsulas aprobadas y publicadas, de acuerdo a una cadena de búsqueda. | Main | * showApprovedCapsulesDescriptionsAndLearnedLessons() |
| GreenSQA | * obtainPublishedCapsulesInfo(String hashtag) |
| KnowledgeCapsule | * getHashtags * getId() * getDescription() * getLearnedLesson() |