

DIAGRAMAS DE REFERENCIA Y DIAGRAMAS GENERADOS

ID	Diagramas de Referencia (Humano)		Diagramas Generados (Herramienta)	
	Código en PlantUML	Diagrama en PlantUML	Código en PlantUML	Diagrama en PlantUML
HU-AC01	<pre> @startuml title Apply for Certified Scrum Practitioner (CSP) Member start :Open "CSP Application"; System :Display application form; Member :Fill required fields; :Attach evidence (files); :Preview application; :Submit; System if (Input valid & files ok?) then (Yes) :Persist application; :Assign trackingId; :Queue confirmation email; >Show success message; stop else (No) >Show inline errors; :Keep entered data; endif @enduml </pre>	<p>Apply for Certified Scrum Practitioner (CSP)</p> <pre> sequenceDiagram actor Member actor System Member->>System: Open "CSP Application" activate System System-->>Member: Display application form Member->>System: Fill required fields Member->>System: Attach evidence (files) Member->>System: Preview application Member->>System: Submit activate System System-->>Member: Input valid & files ok? if Input valid & files ok? then System-->>Member: Persist application System-->>Member: Assign trackingId System-->>Member: Queue confirmation email System-->>Member: Show success message else System-->>Member: Show inline errors System-->>Member: Keep entered data end deactivate System deactivate Member </pre>	<pre> @startuml start :Site member accesses application form; :Site member fills out application; :Site member submits application; if (Application is approved?) then (Yes) :System grants Certified Scrum Practitioner designation; else (No) :System notifies member of rejection; endif stop @enduml </pre>	<pre> stateDiagram-v2 [*] --> SiteMemberAccessesForm SiteMemberAccessesForm --> SiteMemberFillsOut SiteMemberFillsOut --> SiteMemberSubmits SiteMemberSubmits --> Decision{Application is approved?} Decision -- Yes --> SystemGrantsDesignation Decision -- No --> SystemNotifiesRejection SystemGrantsDesignation --> [*] SystemNotifiesRejection --> [*] </pre>

HU-AC02	<pre> @startuml Member start :Open CST application form; :Fill in required information; :Attach supporting documents; :Submit application; System :Show confirmation message; stop @enduml </pre>	<pre> sequenceDiagram participant Member participant System Member->>System: Open CST application form activate System Member->>System: Fill in required information deactivate System Member->>System: Attach supporting documents deactivate System Member->>System: Submit application deactivate System System->>Member: Show confirmation message deactivate Member </pre>	<pre> @startuml start :Site member navigates to the application page; :Site member fills out the application form; :Site member submits the application; :System confirms application submission; stop @enduml </pre>	<pre> graph TD Start(()) --> Nav[Site member navigates to the application page] Nav --> Fill[Site member fills out the application form] Fill --> Submit[Site member submits the application] Submit --> Confirm[System confirms application submission] Confirm --> End((())) </pre>
HU-AC03	<pre> @startuml start :Open list of pending applications; :Select an application; :Review information and documents; if (Application qualifies?) then (Approve) :Approve application; else (Reject) :Reject application; endif stop @enduml </pre>	<pre> sequenceDiagram participant Admin participant System Admin->>System: Open list of pending applications Admin->>System: Select an application Admin->>System: Review information and documents Note over System: Application qualifies? Admin->>System: Approve application Admin->>System: Reject application deactivate Admin deactivate System </pre>	<pre> @startuml start :Site administrator reads practicing and training application; if (Applicant qualifies?) then (Yes) :Site administrator approves application; else (No) :Site administrator rejects application; endif stop @enduml </pre>	<pre> graph TD Start(()) --> Read[Site administrator reads practicing and training application] Read --> Qualifies{Applicant qualifies?} Qualifies -- Yes --> Approve[Site administrator approves application] Qualifies -- No --> Reject[Site administrator rejects application] Approve --> End((())) Reject --> End </pre>

HU-AC04	<pre>@startuml start :Open homepage; :View current news headlines; :Select a news item; :Read news; stop @enduml</pre>	<pre> graph TD Start(()) --> OpenHomepage[Open homepage] OpenHomepage --> ViewHeadlines[View current news headlines] ViewHeadlines --> SelectItem[Select a news item] SelectItem --> ReadNews[Read news] ReadNews --> End((())) </pre>	<pre>@startuml start :Site visitor navigates to home page; :System displays current news; :Site visitor reads current news; stop @enduml</pre>	<pre> graph TD Start(()) --> SiteVisitorNavigates[Site visitor navigates to home page] SiteVisitorNavigates --> SystemDisplays[System displays current news] SystemDisplays --> SiteVisitorReads[Site visitor reads current news] SiteVisitorReads --> End((())) </pre>
HU-AC05	<pre>@startuml start :Open "News Archive" page; :Browse or scroll list of old news; :Select a old news item; :Read old news; stop @enduml</pre>	<pre> graph TD Start(()) --> OpenArchive[Open "News Archive" page] OpenArchive --> BrowseList[Browse or scroll list of old news] BrowseList --> SelectItem[Select a old news item] SelectItem --> ReadOldNews[Read old news] ReadOldNews --> End((())) </pre>	<pre>@startuml start :Site visitor navigates to the old news section; :System displays a list of old news articles; :Visitor selects a news article from the list; :System displays the selected old news article; stop @enduml</pre>	<pre> graph TD Start(()) --> SiteVisitorNavigates[Site visitor navigates to the old news section] SiteVisitorNavigates --> SystemDisplays[System displays a list of old news articles] SystemDisplays --> VisitorSelects[Visitor selects a news article from the list] VisitorSelects --> SystemDisplaysArticle[System displays the selected old news article] SystemDisplaysArticle --> End((())) </pre>

HU-AC06	<pre> @startuml start :Open News page; :Select "Subscribe to RSS"; :Copy or open RSS feed link in reader; stop @enduml </pre>	<pre> graph TD Start(()) --> Open[Open News page] Open --> Select[Select "Subscribe to RSS"] Select --> Copy[Copy or open RSS feed link in reader] Copy --> End((())) </pre>	<pre> @startuml start :Site member navigates to the news section; :Site member finds the RSS feed subscription option; :Site member clicks the subscribe button/link; :System provides the RSS feed URL; :Site member adds the feed URL to an RSS reader; stop @enduml </pre>	<pre> graph TD Start(()) --> SiteMember[Site member navigates to the news section] SiteMember --> Finds[Site member finds the RSS feed subscription option] Finds --> Clicks[Site member clicks the subscribe button/link] Clicks --> Provides[System provides the RSS feed URL] Provides --> Adds[Site member adds the feed URL to an RSS reader] Adds --> End((())) </pre>
---------	---	--	---	--

<p>HU-AC07</p> <pre>@startuml start :Open "Other Courses" page; :View list of upcoming courses; :Select a course of interest; :Read course details; stop @enduml</pre>	<pre> graph TD Start(()) --> Open[Open "Other Courses" page] Open --> View[View list of upcoming courses] View --> Select[Select a course of interest] Select --> Read[Read course details] Read --> End(()) </pre>	<p>@startuml</p> <pre>start :Site visitor navigates to the Other Courses section; :System displays a list of all upcoming Other Courses; :Visitor reviews the list of courses; :Visitor chooses a course; stop @enduml</pre>	<pre> graph TD Start(()) --> SiteVis[Site visitor navigates to the Other Courses section] SiteVis --> SystemDisp[System displays a list of all upcoming Other Courses] SystemDisp --> VisitorReview[Visitor reviews the list of courses] VisitorReview --> VisitorChoose[Visitor chooses a course] VisitorChoose --> End(()) </pre>
--	--	--	--

HU-AC08	<pre> @startuml Trainer start :Open "Create course or event" page; :Enter course or event details; :Save and publish; System :Show confirmation; stop @enduml </pre>	<pre> sequenceDiagram participant Trainer participant System Trainer->>System: Open "Create course or event" page activate Trainer Trainer->>System: Enter course or event details deactivate Trainer Trainer->>System: Save and publish activate System System->>Trainer: Show confirmation deactivate System </pre>	<pre> @startuml start :Trainer initiates creation of a new course or event; if (Selects to create a 'Course')? then (Yes) :Trainer enters course details; :Trainer saves the new course; else (No, creates 'Event') :Trainer enters event details; :Trainer saves the new event; endif :System makes the new course or event visible to site visitors; stop @enduml </pre>	<pre> activityDiagram start(()) --> CreateCourseOrEvent: Trainer initiates creation of a new course or event CreateCourseOrEvent --> SelectCourseEvent: Selects to create a 'Course'? SelectCourseEvent -- Yes --> EnterCourseDetails: Trainer enters course details EnterCourseDetails --> SaveCourse: Trainer saves the new course SaveCourse --> End(()) SelectCourseEvent -- No --> EnterEventDetails: Trainer enters event details EnterEventDetails --> SaveEvent: Trainer saves the new event SaveEvent --> End EnterCourseDetails --> EnterEventDetails EnterCourseDetails --> SaveEvent EnterEventDetails --> SaveCourse EnterEventDetails --> End Note over CreateCourseOrEvent, EnterCourseDetails, EnterEventDetails, SaveCourse, SaveEvent: System makes the new course or event visible to site visitors </pre>
---------	---	---	--	---

HU-AC09	<pre> @startuml Trainer start :Open "My courses or events"; :Select a course or event; >Edit details; :Save changes; System :Show update confirmation; stop @enduml </pre>	<pre> sequenceDiagram participant Trainer participant System Trainer->>System: Open "My courses or events" activate System Trainer->>System: Select a course or event activate System Trainer->>System: Edit details activate System Trainer->>System: Save changes deactivate System System->>Trainer: Show update confirmation deactivate Trainer </pre>	<pre> @startuml start :Trainer selects an existing course or event to update; :System displays the editable details of the course or event; :Trainer modifies the information; :Trainer submits the changes; if (Is the updated information valid?) then (Yes) :System updates the course or event with the new information; :System displays a success confirmation; else (No) :System displays an error message; endif stop @enduml </pre>	<pre> activityDiagram start(()) --> TrainerSelect: Trainer selects an existing course or event to update TrainerSelect --> SystemDisplay: System displays the editable details of the course or event SystemDisplay --> TrainerModify: Trainer modifies the information TrainerModify --> TrainerSubmit: Trainer submits the changes TrainerSubmit --> Decision{Is the updated information valid?} Decision -- Yes --> SystemUpdate: System updates the course or event with the new information SystemUpdate --> Success: System displays a success confirmation Decision -- No --> Error: System displays an error message Success --> End(()) Error --> End </pre>
---------	--	--	--	---

HU-AC10	<pre> @startuml Trainer start :Open "My courses or events"; :Select a course or event; :Delete item; :Confirm deletion; System :Remove from listings; :Show confirmation message; stop @enduml </pre>	<pre> sequenceDiagram participant Trainer participant System Trainer->>System: Open "My courses or events" activate Trainer Trainer->>System: Select a course or event activate Trainer Trainer->>System: Delete item activate Trainer Trainer-->>System: Confirm deletion activate System System-->>Trainer: Remove from listings deactivate System System-->>Trainer: Show confirmation message deactivate System stop </pre>	<pre> @startuml start :Trainer selects a course or event to delete; :Trainer requests deletion; :System asks for confirmation; if (Trainer confirms deletion?) then (Yes) :System deletes the course or event; :System removes the item from the list; :System displays a success message; else (No) :System cancels the deletion process; endif stop @enduml </pre>	<pre> activityDiagram [*] --> Trainer: Trainer selects a course or event to delete Trainer --> System: Trainer requests deletion System --> Decision: System asks for confirmation Decision -- Yes --> System: System deletes the course or event System --> System: System removes the item from the list System --> System: System displays a success message Decision -- No --> System: System cancels the deletion process System --> [*] </pre>
---------	--	---	--	--