

## DIAGRAMAS DE REFERENCIA Y DIAGRAMAS GENERADOS

ID	Diagramas de Referencia		Diagramas Generados	
	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 &amp; files ok?) then (Yes) :Persist application; :Assign trackingId; :Queue confirmation email; &gt;Show success message; stop else (No) &gt;Show inline errors; :Keep entered data; endif @enduml </pre>	<p><b>Apply for Certified Scrum Practitioner (CSP)</b></p> <pre> sequenceDiagram     participant Member     participant System     Member-&gt;&gt;System: Open "CSP Application"     activate System     System-&gt;&gt;Member: Display application form     deactivate System     Member-&gt;&gt;Member: Fill required fields     Member-&gt;&gt;Member: Attach evidence (files)     Member-&gt;&gt;Member: Preview application     Member-&gt;&gt;Member: Submit     activate System     System--&gt;&gt;Member: Input valid &amp; files ok?     Member--&gt;&gt;System: Yes     activate System     System-&gt;&gt;Member: Persist application     System-&gt;&gt;Member: Assign trackingId     System-&gt;&gt;Member: Queue confirmation email     System-&gt;&gt;Member: Show success message     deactivate System     Member--&gt;&gt;System: No     activate System     System-&gt;&gt;Member: Show inline errors     System-&gt;&gt;Member: Keep entered data     deactivate System     end </pre>	<pre> @startuml start :Site member accesses the application form; :Site member fills out the application; :Site member submits the application; if (Application is complete and valid?) then (Yes)     :Grant Certified Scrum Practitioner designation; else (No)     :Notify member of issues with the application; endif stop @enduml </pre>	<pre> stateDiagram-v2     [*] --&gt; SiteMemberAccessesForm     SiteMemberAccessesForm --&gt; SiteMemberFillsOut     SiteMemberFillsOut --&gt; SiteMemberSubmits     SiteMemberSubmits --&gt; Decision{Application is complete and valid?}     Decision --&gt; GrantDesignation     Decision --&gt; NotifyIssues     GrantDesignation --&gt; [*]     NotifyIssues --&gt; [*] </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-&gt;&gt;System: Open CST application form     activate System     Member-&gt;&gt;System: Fill in required information     deactivate System     Member-&gt;&gt;System: Attach supporting documents     deactivate System     Member-&gt;&gt;System: Submit application     deactivate System     System--&gt;&gt;Member: Show confirmation message     </pre>	<pre> @startuml start :Site member navigates to the application form; :Site member fills out the application form; :Site member submits the application; stop @enduml </pre>	<pre> graph TD     Start(( )) --&gt; Nav[Site member navigates to the application form]     Nav --&gt; Fill[Site member fills out the application form]     Fill --&gt; Submit[Site member submits the application]     Submit --&gt; 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 App     Admin-&gt;&gt;App: Open list of pending applications     Admin-&gt;&gt;App: Select an application     Admin-&gt;&gt;App: Review information and documents     Note over App: Application qualifies?     Admin--&gt;&gt;App: Approve application     Admin--&gt;&gt;App: Reject application     </pre>	<pre> @startuml start :Site administrator accesses applications; :Read practicing and training application; if (Applicant qualifies?) then (Yes) :Approve application; else (No) :Reject application; endif stop @enduml </pre>	<pre> graph TD     Start(( )) --&gt; Admin[Site administrator accesses applications]     Admin --&gt; Read[Read practicing and training application]     Read --&gt; Qual{Applicant qualifies?}     Qual -- Yes --&gt; Approve[Approve application]     Qual -- No --&gt; Reject[Reject application]     </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(( )) --&gt; OpenHomepage[Open homepage]     OpenHomepage --&gt; ViewHeadlines[View current news headlines]     ViewHeadlines --&gt; SelectItem[Select a news item]     SelectItem --&gt; ReadNews[Read news]     ReadNews --&gt; End((( )))   </pre>	<pre>@startuml start :Visitor accesses the home page; :System displays the home page; :Visitor reads current news; stop @enduml</pre>	<pre> graph TD     Start(( )) --&gt; VisitorAccesses[Visitor accesses the home page]     VisitorAccesses --&gt; SystemDisplays[System displays the home page]     SystemDisplays --&gt; VisitorReads[Visitor reads current news]     VisitorReads --&gt; 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(( )) --&gt; OpenArchive[Open "News Archive" page]     OpenArchive --&gt; BrowseOld[Browse or scroll list of old news]     BrowseOld --&gt; SelectOld[Select a old news item]     SelectOld --&gt; ReadOld[Read old news]     ReadOld --&gt; End((( )))   </pre>	<pre>@startuml start :Visitor navigates to the old news section; :Visitor browses the list of old news articles; :Visitor selects a news article; :System displays the selected old news article; stop @enduml</pre>	<pre> graph TD     Start(( )) --&gt; VisitorNavigates[Visitor navigates to the old news section]     VisitorNavigates --&gt; VisitorBrowses[Visitor browses the list of old news articles]     VisitorBrowses --&gt; VisitorSelects[Visitor selects a news article]     VisitorSelects --&gt; SystemDisplays[System displays the selected old news article]     SystemDisplays --&gt; 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(( )) --&gt; OpenNews[Open News page]     OpenNews --&gt; SelectRSS[Select "Subscribe to RSS"]     SelectRSS --&gt; CopyReader[Copy or open RSS feed link in reader]     CopyReader --&gt; End((( )))   </pre>	<pre>@startuml start :Site member navigates to the news page; :Site member selects the option to subscribe to the RSS feed; :System presents the RSS feed; stop @enduml</pre>	<pre> graph TD     Start(( )) --&gt; SiteMemberNavigates[Site member navigates to the news page]     SiteMemberNavigates --&gt; SiteMemberSelects[Site member selects the option to subscribe to the RSS feed]     SiteMemberSelects --&gt; SystemPresents[System presents the RSS feed]     SystemPresents --&gt; End((( )))   </pre>

HU-AC07	<pre>@startuml start :Open "Other Courses" page; :View list of upcoming courses; :Select a course of interest; :Read course details; stop @enduml</pre> <pre> sequenceDiagram     participant Visitor     participant System     Visitor-&gt;&gt;System: Open "Other Courses" page     activate System     System-&gt;&gt;Visitor: View list of upcoming courses     deactivate System     Visitor-&gt;&gt;System: Select a course of interest     deactivate System     System-&gt;&gt;Visitor: Read course details     deactivate System     end   </pre>	<pre>@startuml start :Visitor requests to see upcoming Other Courses; :System retrieves the list of Other Courses; if (Are there upcoming Other Courses?) then (Yes)     :System displays the list of upcoming Other Courses;     :Visitor views the list;     :Visitor chooses a course; else (No)     :System displays a message indicating no upcoming courses are available; endif stop @enduml</pre> <pre> sequenceDiagram     participant Visitor     participant System     Visitor-&gt;&gt;System: Visitor requests to see upcoming Other Courses     activate System     System-&gt;&gt;Visitor: System retrieves the list of Other Courses     Note over System: Are there upcoming Other Courses?     if (Yes) then         System-&gt;&gt;Visitor: System displays the list of upcoming Other Courses         Visitor-&gt;&gt;System: Visitor views the list         Visitor-&gt;&gt;System: Visitor chooses a course     else         System-&gt;&gt;Visitor: System displays a message indicating no upcoming courses are available     end     deactivate System     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-&gt;&gt;System: Open "Create course or event" page     Trainer-&gt;&gt;System: Enter course or event details     Trainer-&gt;&gt;System: Save and publish     System-&gt;&gt;Trainer: Show confirmation     deactivate System     end   </pre>	<pre>@startuml start :Trainer initiates creation of a new course or event; :Trainer provides the details for the course or event; :Trainer saves the new course or event; :System creates the new course or event; :System makes the course or event visible to site visitors; stop @enduml</pre> <pre> sequenceDiagram     participant Trainer     participant System     Trainer-&gt;&gt;System: Trainer initiates creation of a new course or event     Trainer-&gt;&gt;System: Trainer provides the details for the course or event     Trainer-&gt;&gt;System: Trainer saves the new course or event     System-&gt;&gt;Trainer: System creates the new course or event     System-&gt;&gt;Trainer: System makes the course or event visible to site visitors     deactivate System     end   </pre>

HU-AC09	<pre> @startuml  Trainer  start :Open "My courses or events"; :Select a course or event; &gt;Edit details; :Save changes;   System  :Show update confirmation; stop @enduml </pre>	<pre> sequenceDiagram     participant Trainer     participant System     Trainer-&gt;&gt;System: Open "My courses or events"     activate System     System-&gt;&gt;Trainer: Select a course or event     Trainer-&gt;&gt;System: Edit details     activate System     System-&gt;&gt;Trainer: Save changes     deactivate System     Trainer--&gt;&gt;System: Show update confirmation     deactivate Trainer     </pre>	<pre> @startuml start :Trainer selects an existing course or event to update; :Trainer modifies the course or event information; :Trainer submits the changes; if (Is the updated information valid?) then (Yes)     :System saves the updated course or event;     :System displays a success message; else (No)     :System displays an error message; endif stop @enduml </pre>	<pre> activityDiagram     start(( )) --&gt; TrainerSelect: Trainer selects an existing course or event to update     TrainerSelect --&gt; TrainerModify: Trainer modifies the course or event information     TrainerModify --&gt; TrainerSubmit: Trainer submits the changes     TrainerSubmit --&gt; Decision{Is the updated information valid?}     Decision -- Yes --&gt; SystemSave: System saves the updated course or event     Decision -- No --&gt; SystemError: System displays an error message     SystemSave --&gt; SystemSuccess: System displays a success message     SystemSuccess --&gt; end(( ))     </pre>
HU-AC10	<pre> @startuml  Trainer  start :Open "My courses or events"; :Select a course or event; &gt;Delete item; :Confirm deletion;   System  :Remove from listings; :Show confirmation message; stop @enduml </pre>	<pre> sequenceDiagram     participant Trainer     participant System     Trainer-&gt;&gt;System: Open "My courses or events"     activate System     System-&gt;&gt;Trainer: Select a course or event     Trainer-&gt;&gt;System: Delete item     activate System     System-&gt;&gt;Trainer: Confirm deletion     deactivate System     Trainer--&gt;&gt;System: Remove from listings     deactivate Trainer     System--&gt;&gt;Trainer: Show confirmation message     deactivate System     </pre>	<pre> @startuml start :Trainer selects a course or event to delete; :Trainer requests to delete the selected item; :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 success message; else (No)     :System cancels the deletion; endif stop @enduml </pre>	<pre> activityDiagram     start(( )) --&gt; TrainerSelect: Trainer selects a course or event to delete     TrainerSelect --&gt; TrainerDelete: Trainer requests to delete the selected item     TrainerDelete --&gt; SystemAsk: System asks for confirmation     SystemAsk --&gt; Decision{Trainer confirms deletion?}     Decision -- Yes --&gt; SystemDelete: System deletes the course or event     Decision -- Yes --&gt; SystemList: System removes the item from the list     Decision -- Yes --&gt; SystemSuccess: System displays success message     Decision -- No --&gt; SystemCancel: System cancels the deletion     SystemDelete --&gt; SystemList     SystemList --&gt; SystemSuccess     SystemSuccess --&gt; end(( ))     </pre>