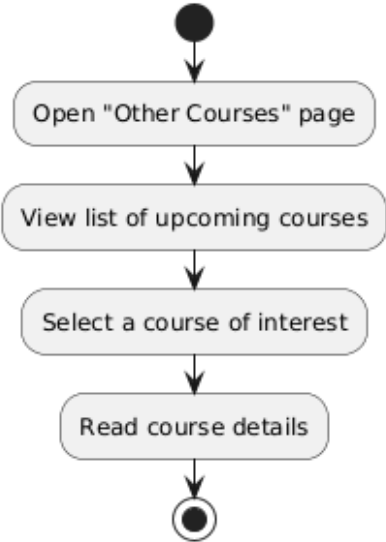
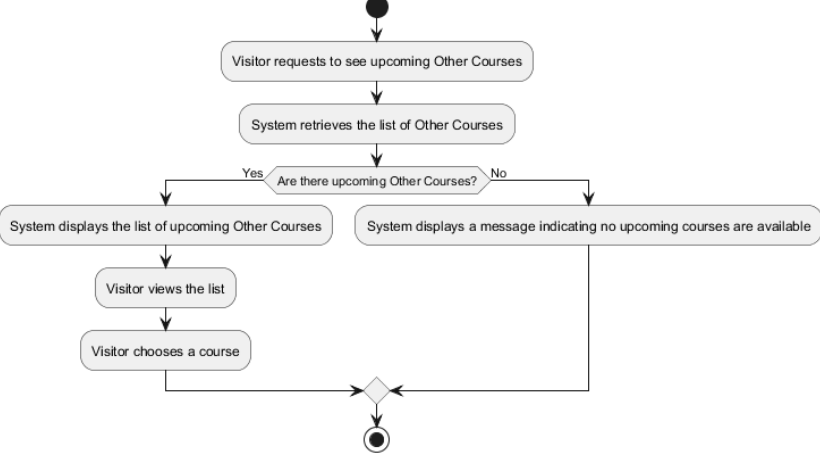
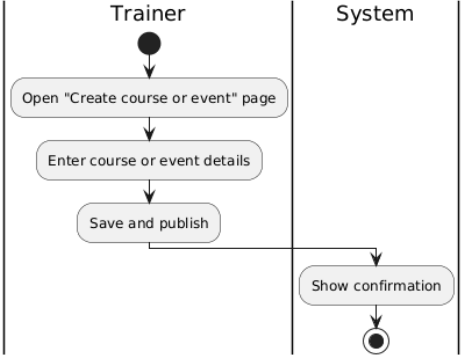



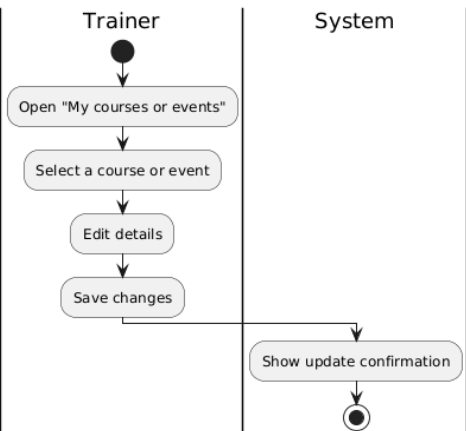
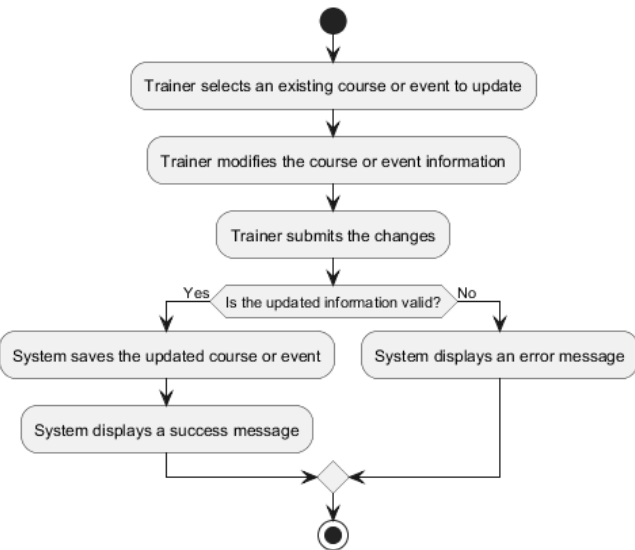
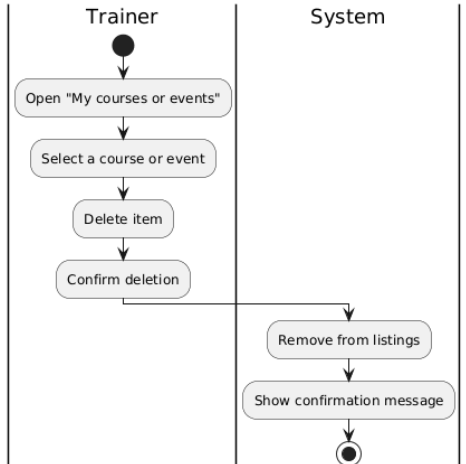
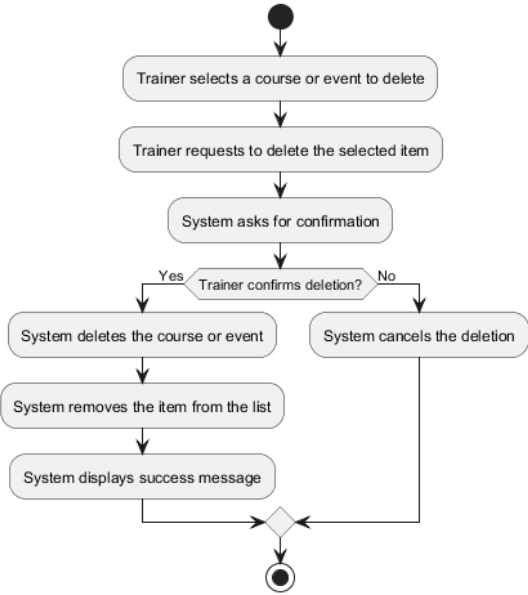
DIAGRAMAS DE REFERENCIA Y DIAGRAMAS GENERADOS

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 & 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>The diagram shows a Member actor and a System boundary. The Member starts with 'Open "CSP Application"', which triggers the System to 'Display application form'. The Member then performs 'Fill required fields', 'Attach evidence (files)', and 'Preview application', leading to a 'Submit' use case. This triggers a decision 'Input valid & files ok?'. If 'Yes', the System performs 'Persist application', 'Assign trackingId', 'Queue confirmation email', and 'Show success message' before ending. If 'No', the System performs 'Show inline errors' and 'Keep entered data'.</p>	<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>	<p>The activity starts with a start node leading to 'Site member accesses the application form', followed by 'Site member fills out the application' and 'Site member submits the application'. A decision node 'Application is complete and valid?' follows. The 'Yes' path leads to 'Grant Certified Scrum Practitioner designation', and the 'No' path leads to 'Notify member of issues with the application'. Both paths merge at a join node before reaching the end node.</p>

HU-AC02	<p>@startuml [Member] start :Open CST application form; :Fill in required information; :Attach supporting documents; :Submit application; [System] :Show confirmation message; stop @enduml</p>	<pre>sequenceDiagram participant Member participant System Member->>Member: Open CST application form Member->>Member: Fill in required information Member->>Member: Attach supporting documents Member->>Member: Submit application Member->>System: System->>System: Show confirmation message</pre>	<p>@startuml start :Site member navigates to the application form; :Site member fills out the application form; :Site member submits the application; stop @enduml</p>	<pre>graph TD Start(()) --> A[Site member navigates to the application form] A --> B[Site member fills out the application form] B --> C[Site member submits the application] C --> End((()))</pre>
HU-AC03	<p>@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</p>	<pre>graph TD Start(()) --> A[Open list of pending applications] A --> B[Select an application] B --> C[Review information and documents] C --> D{Application qualifies?} D -- Approve --> E[Approve application] D -- Reject --> F[Reject application] E --> G{ } F --> G G --> End((()))</pre>	<p>@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</p>	<pre>graph TD Start(()) --> A[Site administrator accesses applications] A --> B[Read practicing and training application] B --> C{Applicant qualifies?} C -- Yes --> D[Approve application] C -- No --> E[Reject application] D --> F{ } E --> F F --> End((()))</pre>

HU-AC04	<p>@startuml start :Open homepage; :View current news headlines; :Select a news item; :Read news; stop @enduml</p>	<pre>graph TD; Start(()) --> UC1([Open homepage]); UC1 --> UC2([View current news headlines]); UC2 --> UC3([Select a news item]); UC3 --> UC4([Read news]); UC4 --> End((()))</pre>	<p>@startuml start :Visitor accesses the home page; :System displays the home page; :Visitor reads current news; stop @enduml</p>	<pre>graph TD; Start(()) --> UC1([Visitor accesses the home page]); UC1 --> UC2([System displays the home page]); UC2 --> UC3([Visitor reads current news]); UC3 --> End((()))</pre>
HU-AC05	<p>@startuml start :Open "News Archive" page; :Browse or scroll list of old news; :Select a old news item; :Read old news; stop @enduml</p>	<pre>graph TD; Start(()) --> UC1([Open "News Archive" page]); UC1 --> UC2([Browse or scroll list of old news]); UC2 --> UC3([Select a old news item]); UC3 --> UC4([Read old news]); UC4 --> End((()))</pre>	<p>@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</p>	<pre>graph TD; Start(()) --> UC1([Visitor navigates to the old news section]); UC1 --> UC2([Visitor browses the list of old news articles]); UC2 --> UC3([Visitor selects a news article]); UC3 --> UC4([System displays the selected old news article]); UC4 --> End((()))</pre>
HU-AC06	<p>@startuml start :Open News page; :Select "Subscribe to RSS"; :Copy or open RSS feed link in reader; stop @enduml</p>	<pre>graph TD; Start(()) --> UC1([Open News page]); UC1 --> UC2([Select "Subscribe to RSS"]); UC2 --> UC3([Copy or open RSS feed link in reader]); UC3 --> End((()))</pre>	<p>@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</p>	<pre>graph TD; Start(()) --> UC1([Site member navigates to the news page]); UC1 --> UC2([Site member selects the option to subscribe to the RSS feed]); UC2 --> UC3([System presents the RSS feed]); UC3 --> End((()))</pre>

HU-AC07	<p>@startuml start :Open "Other Courses" page; :View list of upcoming courses; :Select a course of interest; :Read course details; stop @enduml</p>	 <pre>graph TD; Start(()) --> UC1[Open "Other Courses" page]; UC1 --> UC2[View list of upcoming courses]; UC2 --> UC3[Select a course of interest]; UC3 --> UC4[Read course details]; UC4 --> End((()))</pre>	<p>@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</p>	 <pre>graph TD; Start(()) --> UC1[Visitor requests to see upcoming Other Courses]; UC1 --> UC2[System retrieves the list of Other Courses]; UC2 --> Decision{Are there upcoming Other Courses?}; Decision -- Yes --> UC3[System displays the list of upcoming Other Courses]; UC3 --> UC4[Visitor views the list]; UC4 --> UC5[Visitor chooses a course]; UC5 --> Join(()); Decision -- No --> UC6[System displays a message indicating no upcoming courses are available]; UC6 --> Join; Join --> End((()))</pre>
HU-AC08	<p>@startuml Trainer start :Open "Create course or event" page; :Enter course or event details; :Save and publish; System :Show confirmation; stop @enduml</p>	 <pre>sequenceDiagram; participant Trainer; participant System; Trainer->>System: Open "Create course or event" page; Trainer->>System: Enter course or event details; Trainer->>System: Save and publish; System->>Trainer: Show confirmation; Trainer->>System: ;</pre>	<p>@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</p>	 <pre>graph TD; Start(()) --> UC1[Trainer initiates creation of a new course or event]; UC1 --> UC2[Trainer provides the details for the course or event]; UC2 --> UC3[Trainer saves the new course or event]; UC3 --> UC4[System creates the new course or event]; UC4 --> UC5[System makes the course or event visible to site visitors]; UC5 --> End((()))</pre>

HU-AC09	<p>@startuml [Trainer] start :Open "My courses or events"; :Select a course or event; :Edit details; :Save changes; [System] :Show update confirmation; stop @enduml</p>	 <p>The diagram shows two swimlanes: Trainer and System. In the Trainer swimlane, the process starts with 'Open "My courses or events"', followed by 'Select a course or event', 'Edit details', and 'Save changes'. An arrow from 'Save changes' points to the System swimlane, where the process continues with 'Show update confirmation' and ends at a final node.</p>	<p>@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</p>	 <p>The diagram shows a single swimlane. It starts with 'Trainer selects an existing course or event to update', followed by 'Trainer modifies the course or event information', and 'Trainer submits the changes'. A decision diamond 'Is the updated information valid?' follows. The 'Yes' path leads to 'System saves the updated course or event' and then 'System displays a success message'. The 'No' path leads to 'System displays an error message'. Both paths merge at a final node.</p>
HU-AC10	<p>@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</p>	 <p>The diagram shows two swimlanes: Trainer and System. In the Trainer swimlane, the process starts with 'Open "My courses or events"', followed by 'Select a course or event', 'Delete item', and 'Confirm deletion'. An arrow from 'Confirm deletion' points to the System swimlane, where the process continues with 'Remove from listings' and 'Show confirmation message', ending at a final node.</p>	<p>@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</p>	 <p>The diagram shows a single swimlane. It starts with 'Trainer selects a course or event to delete', followed by 'Trainer requests to delete the selected item', and 'System asks for confirmation'. A decision diamond 'Trainer confirms deletion?' follows. The 'Yes' path leads to 'System deletes the course or event', 'System removes the item from the list', and 'System displays success message'. The 'No' path leads to 'System cancels the deletion'. Both paths merge at a final node.</p>