

# Rúbricas de Evaluación Por Sprint

## Módulo Programador Web 2023

<u>CRITERIO</u>	<u>LOGRADO</u>	<u>EN PROCESO</u>	<u>NO LOGRADO</u>	<u>NO EVIDENCIA</u>
<i>Rúbricas de valoración Sprint 0</i>				
Logra conectarse al repositorio GitHub (colaboradores)				
Logra agregar Issues como Historia de Usuarios, respetando la nomenclatura sugerida.				
Logra crear Milestone por Sprints.				
Logra crear Wiki con registro de ceremonias por SPRINT (daily, review, retrospective) y novedades del equipo.				
Logra documentar el proyecto mediante en IEEE 830 para el cliente: más requerimientos.				
Utiliza kanban (Product Backlog, ToDo, In process, Finished).				

Logra crear diagramas casos de uso				
Logra crear diagramas de clase				
Logra mejorar el modelo de datos lógico en base a los nuevos requerimientos (e-commerce)				
<i>Rúbricas de valoración Sprint 1</i>				
Logra crear SPA con módulos, componentes y routing en Angular (HOME, REGISTRO, LOGIN, DASHBOARD, PRODUCTO o SERVICIOS, entre otros)				
Logra implementar las clases en Python en base al diagrama de clases				
Logra realizar Commit: queda registro de participación de todos los miembros del grupo.				
Logra crear una versión Draft del código fuente (Git y GitHub)				
Logra mantener actualizado el Kanban del proyecto				

Logra mantener actualizado el doc IEEE830				
Logra mantener actualizado el flujo de las Issues				
<i>Rúbricas de valoración Sprint 2</i>				
Logra implementar formularios reactivos				
Logra crear servicios en Angular que realicen peticiones HTTP.				
Logra implementar DataBinding				
Logra crear un API Rest para obtención de datos.				
Logra realizar Commit: queda registro de participación de todos los miembros del grupo.				
Logra crear una versión Draft del código fuente (Git y GitHub)				
Logra mantener actualizado el Kanban del proyecto				
Logra mantener actualizado el doc IEEE830				
Logra mantener				

actualizado el flujo de las Issues				
<i>Rúbricas de valoración Sprint 3</i>				
Logra conectar los servicios en Angular con las API Rest implementadas en Django				
Logra incorporar Login funcional con consumo de datos.				
Logra la comunicación entre componentes				
Logra implementar API con funcionalidad para login +JWT				
Logra crear CRUD en SQL: select- insert - update - delete - JOIN.				
Logra realizar Commit: queda registro de participación de todos los miembros del grupo.				
Logra crear una versión Draft del código fuente (Git y GitHub)				
Logra mantener actualizado el Kanban del proyecto				
Logra mantener				

actualizado el doc IEEE830				
Logra mantener actualizado el flujo de las Issues				
<i>Rúbricas de valoración Sprint 4</i>				
Logra una aplicacion funcional completa con JWT implementado desde Front				
Logra implementar el manejo de errores en Servicios (Frontend).				
Logra implementar el manejo de errores (Backend).				
Logra realizar Commit: queda registro de participación de todos los miembros del grupo.				
Logra crear una versión <b>Release</b> del código fuente (Git y GitHub)				
Logra generar un archivo Readme.md				
Logra mantener				

actualizado el Kanban del proyecto				
Logra completar el doc IEEE830				
Logra mantener actualizado el flujo de las Issues				