

**PROPUESTA MODULO DE TALLER** (para aprobación por la Comisión de Carrera)

|                             |   |
|-----------------------------|---|
| Nombre Actividad Específica | <i>Implementación del metamodelo Object-Centric Event Data (OCED) para minería de procesos como biblioteca Python</i>   |
| Proponente                  | <i>Instituto de Computación</i>   |
| Responsable                 | <i>Daniel Calegari</i>  |
| Responsable en INCO o FING  | <i>Daniel Calegari (<a href="mailto:dcalegar@fing.edu.uy">dcalegar@fing.edu.uy</a>)</i>   |
| Objetivo                    | <i>El objetivo general de esta actividad es extender una implementación base realizada del metamodelo Object-Centric Event Data (OCED) para minería de procesos como biblioteca Python con desarrollo dirigido por modelos(MDE).</i>  |
| Descripción                 | <p><i>La Minería de Procesos (PM, [1]) permite analizar eventos de tiempo de ejecución de un sistema de información para, entre otras cosas, descubrir los procesos de negocio correspondientes. Para ello, necesita registrar en un log cada evento ocurrido y la fecha de ocurrencia, entre otros datos de interés. Recientemente se comenzó a trabajar en lo que se conoce como object-centric process mining (OCPM) que permite vincular a los eventos con diversos objetos (datos organizacionales) y realizar el análisis desde la perspectiva de dichos objetos. Se está definiendo el estándar OCED [2] que permite representar un log siguiendo la idea de OCPM y, recientemente, se definió un metamodelo para representación de un log OCPM y herramientas para su manipulación en Java [3].</i></p> <p><i>Existen diversas herramientas que brindan soporte a la minería de procesos, las cuales se encuentran actualmente comenzando a soportar el enfoque OCPM. En particular, pm4py [4] es una herramienta en Python muy utilizada por la comunidad. Dada la existencia del metamodelo OCED, aunque en el entorno Java, parece razonable buscar compatibilizar ambas propuestas en el entorno Python. Para ello, se podría utilizar una biblioteca de representación de metamodelos Ecore en Python como PyEcore [5].</i></p> <p><i>Actualmente existe una primera implementación de la biblioteca OCED en Python, aunque es necesaria su adaptación a un cambio en el metamodelo de referencia, así como organizar su empaquetado y difusión, generación de materiales e integración con otras propuestas existentes.</i></p> <p><i>El alcance concreto del trabajo implica:</i></p> <ol style="list-style-type: none"> <li><i>1. Adaptación de la biblioteca al nuevo formato OCED</i></li> <li><i>2. Empaquetado de la biblioteca para su distribución</i></li> <li><i>3. Elaboración de sitio web de OpenOCED para publicación de las bibliotecas Java y Python, junto con ejemplos y otras herramientas asociadas</i></li> <li><i>4. Interacción de la biblioteca con los formatos OCEL 2.0 [6] y OCED-PG [7] a través de transformaciones</i></li> <li><i>5. Generación de ejemplos que conformen con el metamodelo de OCED a partir de los ejemplos existentes de OCEL 2.0 y OCED-PG</i></li> </ol> <p><i>[1] W. van der Aalst, Process Mining: Data Science in Action 2<sup>nd</sup> ed., (2016)</i><br/> <i>[2] Call for Action: OCED Reference Implementations</i></p> |

|  |   |
|--|---|
|  | <a href="https://www.tf-pm.org/news/call-for-action-oced-reference-implementations">https://www.tf-pm.org/news/call-for-action-oced-reference-implementations</a><br>[3] OCED metamodel. <a href="https://gitlab.fing.edu.uy/open-coal/oced/">https://gitlab.fing.edu.uy/open-coal/oced/</a><br>[4] pm4py. <a href="https://pm4py.fit.fraunhofer.de/">https://pm4py.fit.fraunhofer.de/</a><br>[5] PyEcore. <a href="https://github.com/pyecore/pyecore">https://github.com/pyecore/pyecore</a><br>[6] OCEL. <a href="https://ocel-standard.org/">https://ocel-standard.org/</a><br>[7] OCED-PG. <a href="https://zenodo.org/records/8296559">https://zenodo.org/records/8296559</a> |
| Aporte a / tareas concretas del estudiante | <i>Los estudiantes profundizarán sus conocimientos sobre aplicación de MDE y desarrollo web.</i>  |
|  |   |
| Carga horaria total                        | <i>150 horas (10 créditos)</i>  |
| Carga horaria sem.                         | <i>10 horas</i>   |
| Fecha inicio                               | <i>Diciembre de 2023</i>  |
| Fecha fin                                  | <i>Marzo de 2024</i>  |
| Conocimientos requeridos                   | <i>Conocimientos de Ingeniería Dirigida por Modelos.</i>  |
| Cupo de estudiantes                        | <i>2-3 (debido a la carga del trabajo a realizar)</i>   |
| Forma de Selección                         | <i>Existe un grupo de estudiante del curso TMDE que trabajó en la definición del metamodelo en Python.</i>  |
| Método de Evaluación                       | <i>Entrega de las bibliotecas y código asociado funcionando según escenarios de prueba definidos, informe del trabajo, manual técnico</i>   |

Firma docente responsable  
inco – fing

aprobado Comisión Carrera fecha: