

Córdoba, 23/10/2017

VISTO: El proyecto de Capacitación: **Postítulo de Especialización docente de nivel superior en enseñanza de la programación informática** registrado en la R.P.F.D.C. bajo el Número: 1 33-3 de 400 horas Expdte. N°: S/N

de Modalidad **Semi Presencial**

cuyos destinatarios son **Docentes nivel: Primario, Secundario y Superior**

presentado por la Entidad Oferente: **Instituto Superior de Formación Docente Simón Bolívar**

La definición de las competencias correspondientes a cada Ministerio establecidas por la Ley Orgánica N° 9454,

Y CONSIDERANDO:

Que la Entidad Oferente se encuentra debidamente registrada y no posee inhibiciones o sanciones de ningún tipo,

Que el proyecto se presentó en tiempo y forma, siendo evaluado favorablemente por el Equipo Técnico de Evaluación cuyo informe consta en los actuados motivo del presente,

Que lo solicitado se adecua a las previsiones del Decreto N° 1605/ 2003 y las Resoluciones Ministeriales N° 1506/ 2003 N° 1490/ 2003, N° 67/ 2005, N° 76/ 2007, N° 523/ 2008 y N° 398/ 2011,

Que compete a la Subsecretaría de Estado de Promoción de la Igualdad y Calidad Educativa entender en lo concerniente a la ejecución de los planes, programas y proyectos de su área, y al ejercicio del poder de policía en todo el territorio provincial,

Que la Subsecretaría de Estado de Promoción de la Igualdad y Calidad Educativa se avoca al conocimiento y resolución de los asuntos que corresponden a los fines y objetivos de la Formación Docente en el ámbito Provincial,

Por ello, Conforme al Artículo 24 del Decreto 2174/ 07 ratificado por Ley 9454,

El Subsecretario de Estado de Promoción de la Igualdad y Calidad Educativa
del Ministerio de Educación de la Provincia de Córdoba

RESUELVE

ART.1º: ACEPTAR el Proyecto: **Postítulo de Especialización docente de nivel superior en enseñanza de la programación informática**

de **400 horas reloj**

presentado por el Oferente **Instituto Superior de Formación Docente Simón Bolívar**

de Modalidad **Semi Presencial**

cuyos destinatarios son **Docentes nivel: Primario, Secundario y Superior**

Sede: 1 / 2017 Departamento: Capital

Localidad: CORDOBA

Agustín Garzón N° 1225

Capacitador/es:

11111111	Wolovich	Nicolás
17001542	Fridlender	Daniel
17918440	Arroyo	Daniel
18053307	Cuello	Silvia
20555523	Blanco	Javier
23983484	Novaira	María Marta
26082986	Martinez	María Cecilia
29074461	Rodriguez	Eduardo S
29202576	Onetti	Carlos
29224785	Gaitán	Martin Emilio
31126809	Dipierrí	Iris Carolina

Cronograma:

Soledad Ana Martín
Prof. Soledad Ana Martín
Evaluación de Proyectos y Monitoreo
Instituto Superior de Formación Docente Continua

RESOLUCIÓN DE ACEPTACIÓN N° 221/2017

33535532 Gómez Marcos

Encuentro Presencial: 10/03/2018	9:0	15:0	Primer Encuentro
Encuentro Presencial: 17/03/2018	9:0	15:0	
Encuentro Presencial: 07/04/2018	9:0	15:0	
Encuentro Presencial: 21/04/2018	9:0	15:0	
Encuentro Presencial: 21/04/2018	14:0	19:0	Seminario Trabajo Final I
Encuentro Presencial: 05/05/2018	9:0	15:0	
Encuentro Presencial: 19/05/2018	9:0	15:0	
Encuentro Presencial: 02/06/2018	9:0	15:0	
Encuentro Presencial: 16/06/2018	9:0	15:0	
Encuentro Presencial: 16/06/2018	14:0	19:0	Seminario Trabajo Final I
Encuentro Presencial: 30/06/2018	9:0	15:0	Encuentro de Evaluación
Encuentro Presencial: 04/08/2018	9:0	15:0	
Encuentro Presencial: 18/08/2018	9:0	15:0	
Encuentro Presencial: 01/09/2018	9:0	15:0	
Encuentro Presencial: 15/09/2018	9:0	15:0	
Encuentro Presencial: 15/09/2018	14:0	19:0	Seminario Trabajo Final I
Encuentro Presencial: 06/10/2018	9:0	15:0	
Encuentro Presencial: 20/10/2018	9:0	15:0	
Encuentro Presencial: 03/11/2018	9:0	15:0	
Encuentro Presencial: 10/11/2018	9:0	15:0	
Encuentro Presencial: 10/11/2018	14:0	19:0	Seminario Trabajo Final I
Encuentro Presencial: 24/11/2018	9:0	15:0	Encuentro de Evaluación
Encuentro Presencial: 09/03/2019	9:0	15:0	
Encuentro Presencial: 23/03/2019	9:0	15:0	
Encuentro Presencial: 06/04/2019	9:0	15:0	
Encuentro Presencial: 20/04/2019	9:0	15:0	
Encuentro Presencial: 20/04/2019	14:0	19:0	Seminario Trabajo Final II
Encuentro Presencial: 04/05/2019	9:0	15:0	
Encuentro Presencial: 18/05/2019	9:0	15:0	
Encuentro Presencial: 01/06/2019	9:0	15:0	
Encuentro Presencial: 15/06/2019	9:0	15:0	
Encuentro Presencial: 15/06/2019	14:0	19:0	Seminario Trabajo Final II
Encuentro Presencial: 29/06/2019	9:0	15:0	Encuentro de Evaluación
Encuentro Presencial: 03/08/2019	9:0	15:0	
Encuentro Presencial: 17/08/2019	9:0	15:0	
Encuentro Presencial: 31/08/2019	9:0	15:0	
Encuentro Presencial: 14/09/2019	9:0	15:0	
Encuentro Presencial: 14/09/2019	14:0	19:0	Seminario Trabajo Final II
Encuentro Presencial: 05/10/2019	9:0	15:0	
Encuentro Presencial: 19/10/2019	9:0	15:0	
Encuentro Presencial: 02/11/2019	9:0	15:0	
Encuentro Presencial: 16/11/2019	9:0	15:0	
Encuentro Presencial: 16/11/2019	14:0	19:0	Seminario Trabajo Final II
Encuentro Presencial: 23/11/2019	9:0	15:0	Encuentro de Evaluación

ART. 3º: Protocolícese, notifíquese y dese copia.-

RESOLUCIÓN DE ACEPTACIÓN N°: 221/2017



Handwritten signature of Dr. Horacio A. Ferreyra

Dr. Horacio A. Ferreyra
 Subsecretario de Estado de Promoción de
 Igualdad y Calidad Educativa

J.D.

Handwritten notes:
 Cuenta de
 Franco N.E.H
 06/11/2017

ES COPIA FIEL DEL ORIGINAL

Handwritten signature of Soledad Ana Martín
 Soledad Ana Martín
 Oficina de Promoción de Proyectos y Monitoreo
 Subsecretaría de Formación Docente Continua

Especialización Docente de Nivel Superior en Enseñanza de la Programación en Informática

La Facultad de Matemática, Astronomía, Física y Computación de la Universidad Nacional de Córdoba (FAMAF) en forma conjunta con el Ministerio de Educación de la Provincia de Córdoba, a través del Instituto de Enseñanza Superior “Simón Bolívar”, ofrece a los docentes de Educación Primaria, Secundaria y Superior la Especialización Docente de Nivel Superior en Enseñanza de la Programación en Informática.

Esta iniciativa puede leerse en consonancia con la Res. 263/15 del Consejo Federal de Educación que destaca la relevancia que reviste en la actualidad la enseñanza y el aprendizaje significativo de la programación (...); “la Programación es de importancia estratégica en el Sistema Educativo Nacional durante la escolaridad obligatoria, para fortalecer el desarrollo económico y social de la Nación, conforme lo establecido por el artículo 3o de la Ley de Educación Nacional.” (art. 1)

En este marco, la enseñanza de la programación se plantea como una estrategia clave en la construcción de una ciudadanía alfabetizada en los lenguajes de nuestro tiempo. La enseñanza de la programación en la escuela promueve herramientas para comprender las lógicas de funcionamiento de la tecnología digital. Es decir, se trata de reconocer el conjunto de decisiones que toman y los saberes que dominan quienes desarrollan el software que controla dichas tecnologías. El conocimiento de estos saberes alienta un modo de pensamiento afín a los desafíos de una sociedad atravesada por las nuevas tecnologías. Al decir de Palamidessi (2014, p. 3), “... la tecnología nunca es reductible a un problema técnico (...) El problema se complica cuando el establishment educativo tiende a considerar a la tecnología como un objeto que no es digno de reflexión; un simple medio. Pero se trata de un error fatal”.

La Especialización tiene como objetivo formar a los docentes en las bases de la programación y su didáctica. El plan de estudios está organizado en proyectos acotados de programación siguiendo los principios del aprendizaje por descubrimiento, enfoque pertinente y privilegiado para su didáctica.

Los proyectos seleccionados tienen la intencionalidad de posibilitar la integración de contenidos fundamentales de la programación con los saberes pedagógicos y tecnológicos necesarios para su enseñanza. En ese sentido, se ofrecerá una formación que integre contenidos de la disciplina con saberes de la didáctica y pedagogía.

Los contenidos se presentan en un currículum de formato espiralado al modo de nociones que van ampliando su dificultad en tres niveles (desde la familiaridad al uso y, luego, al dominio) a lo largo de la especialidad. De este modo, se espera que los docentes, en primera instancia, se familiaricen con algunos conceptos; luego, los usen en un programa o análisis y, finalmente, produzcan nuevos saberes a partir del dominio conceptual.

En síntesis, la propuesta tiene el objetivo de formar a los docentes en la enseñanza de una área relativamente nueva para las instituciones educativas, recuperando “buenas prácticas” de formación

docente: currículum por proyectos, saberes disciplinares y didácticos integrados, y contenidos espiralados.

Título a otorgar

Especialista Docente de Nivel Superior en Enseñanza de la Programación en Informática

Duración

Dos años

Carga horaria

400 horas

Destinatarios

Docentes de Educación Primaria, Educación Secundaria y Educación Superior.

Requisitos de admisión

Es necesario tener formación docente para acceder a la titulación de especialista docente.

Se incentiva la inscripción de tres docentes por institución para promover el diseño e implementación de proyectos situados y la colaboración entre colegas. Si bien la admisión es abierta, en caso de cubrir el cupo, se le dará prioridad a aquellos docentes que se hayan inscripto en conjunto con otros docentes de la misma escuela y a docentes en ejercicio.

Perfil de los Egresados

Los docentes egresados de esta especialización tendrán un dominio de conceptos fundamentales de la programación que les permitirá seleccionar plataformas diseñadas para la enseñanza de la programación y contenidos disciplinares para enseñar a sus estudiantes, organizar una propuesta curricular, llevar a cabo experiencias de enseñanza de conceptos de programación, particularmente, basadas en proyectos en los niveles primario y secundario.

Asimismo, podrán brindar asesoramiento a instituciones de educación formal y no formal sobre contenidos y estrategias para enseñar conceptos de programación en el nivel primario y secundario.

Objetivos Generales

Que los docentes sean capaces de:

- Desarrollar programas de computación sencillos a partir del dominio de un lenguaje de programación.

- Comprender, seleccionar y aplicar los conceptos fundamentales de la programación en el desarrollo de proyectos educativos en los diferentes niveles educativos.
- Diseñar experiencias para la enseñanza los conceptos fundamentales de la programación.
- Diseñar experiencias de enseñanza que requieran recuperar los modos de pensamiento propios de la programación.
- Diseñar experiencias de enseñanza que permitan un abordaje crítico de las tecnologías informáticas y su relación con la sociedad.
- Promover trabajos colaborativos entre equipos docentes a partir de las posibilidades que ofrecen los proyectos vinculados a la programación informática.

Objetivos Específicos

- Que los docentes se apropien de estrategias para diseñar proyectos integradores en sus escuelas que se articulen con la enseñanza de la programación.
- Que los docentes incorporen a su práctica estrategias de enseñanza por descubrimiento y de competencias de alto orden tales como resolución de problemas, identificación de patrones y abstracción.
- Comprender el funcionamiento de una computadora en tanto mecanismo que computa, independientemente de cualquier plataforma o dispositivo. Comprender la noción de máquina universal.
- Entender los mecanismos de funcionamiento de una computadora como un transformador de estados a partir de la ejecución de instrucciones simples, reforzando la relación entre hardware y software.
- Comprender cómo operan las nociones centrales de los lenguajes: procedimientos, funciones, variables, asignación, constructor secuencial, constructor de alternativa, constructor de repetición, etc.
- Que a partir de comprender cómo funciona una computadora, los docentes puedan adquirir seguridad para modificar software y hardware sin el temor al rompimiento y promover en los alumnos el uso de repositorios abiertos y apropiación de los dispositivos físicos.
- Que los docentes conozcan los conceptos básicos de administración y configuración de redes y sistemas operativos y que sean capaces de instalar y administrar software educativo.
- Que a partir del dominio de un lenguaje que haga las veces de meta-herramienta (por ejemplo Python) los docentes puedan ganar seguridad en términos de su aplicación para apropiarse y comprender otros lenguajes de programación de base imperativa.
- Que los docentes puedan desarrollar estrategias y heurísticas simples de resolución de problemas: lógicos, algorítmicos y de formalizar su resolución.
- Que los docentes puedan desarrollar un programa sencillo con entrada salida.
- Que el docente pueda abordar la enseñanza de los conceptos fundamentales de la computación en los diferentes niveles del sistema educativo obligatorio.

- Proveer al docente de herramientas conceptuales que le permitan discutir con sus alumnos sobre las implicancias sociales en el diseño de los objetos técnicos, y de la computación en particular, y su utilización.
- Contribuir al debate sobre el uso de software libre en el ámbito de la enseñanza y las instituciones públicas.
- Discutir sobre las bases filosóficas de la protección legal de los derechos a la privacidad. Que puedan analizar conceptualmente nuevas tecnologías en términos de sus licencias y mecanismos de recolección, almacenamiento y análisis de datos personales. (ej, cámaras de seguridad, manipulación de información de contactos o actividades, etc).
- Que puedan discutir sobre propiedad intelectual y licencias. Analizar las implicancias sobre la apropiación del conocimiento y el modo de construcción de software que permiten las licencias libres. Reflexionar sobre el uso de este tipo de licencias aplicadas a la producción artística.

Carga horaria de la carrera

Módulo	Formato curricular	Cant .Hor as
PRIMER AÑO		
Lógica, Programación y su Enseñanza	Proyecto	40
Ensamble y Programación de un Robot	Proyecto	40
Administración y Configuración de Herramientas Informáticas	Proyecto	40
Introducción a lenguajes de programación a través de animaciones y videojuegos	Proyecto	40
Seminario de Trabajo Final I	Seminario	40
SEGUNDO AÑO		
Resolución de problemas de programación a partir del diseño de juegos interactivos	Proyecto	40
Introducción a la programación orientada a objetos a partir del desarrollo de un sistema de votación	Proyecto	40
Procesamiento de datos a partir de análisis de bases de datos públicas	Proyecto	40
Desarrollo de Aplicaciones Web	Proyecto	40
Seminario de Trabajo Final II	Seminario	40

Régimen académico específico

La propuesta consta de cuatrocientas horas de cursado distribuidas en dos años.

El plan de estudios está conformado por diez módulos, ocho de ellos con formato de proyecto y dos seminarios de trabajo final. El diseño de cada módulo y la coordinación académica del mismo estará a cargo de uno o dos docentes responsables. Los cursantes se distribuirán en cursos (aulas) de aproximadamente 25 personas y tendrán un docente asignado (tutor) encargado del dictado de las clases presenciales por aula, tutorías, el seguimiento de las clases y la evaluación.

Cada módulo-proyecto de 40 horas cuenta con 2 (dos) clases virtuales de 8hs, 4 (cuatro) clases presenciales de 5hs y un encuentro de cierre de 4hs. Durante las horas de trabajo a distancia se promoverán actividades virtuales y experiencias para que los docentes realicen con sus colegas en escuelas y en aulas con estudiantes. Los encuentros presenciales se realizarán los días sábados, con reuniones quincenales en donde se dictarán clases teórico-prácticas de 8:30 a 13:30hs con el docente responsable o el tutor, según corresponda.

Los seminarios de trabajo final se estructuran a lo largo de la carrera con encuentros presenciales y/o tutorías para acompañar el análisis, diseño, implementación en el aula y evaluación de las propuestas de enseñanza de la programación que definan los cursantes en el marco de lo trabajado en el resto de los módulos.

Antecedentes académicos y profesionales del plantel docente

Los docentes responsables de elaborar los módulos tienen diversas experiencias de formación. El plantel incluye a investigadores de CONICET del área de Computación y Educación, Doctores de Computación y profesores de Universidades Nacionales, Licenciados en Computación y Profesores de Matemáticas con experiencia docente en niveles primarios, secundario, terciario y universitario. Asimismo, la mitad del equipo ha diseñado, ejecutado y estudiado experiencias de formación docente continua. Esta variedad de trayectorias formativas permite elaborar un programa académico donde se incorporan diferentes dimensiones y perspectivas.

Descripción de cada módulo

Lógica, Programación y su Enseñanza

Justificación

Este módulo recupera y revisa ideas previas sobre la computación y su enseñanza. Además, introduce a los docentes en la lógica de la programación a través de juegos lógicos y situaciones

problemáticas sin el uso de la computadora.

Articulación con la práctica de la enseñanza: En este módulo se analiza la introducción de la programación en la escuela en tanto innovación educativa y se aborda su relación con la calidad educativa. Se abordan diferentes concepciones de pensamiento computacional.

Objetivos

- Fomentar estrategias de enseñanza por descubrimiento y competencias tales como resolución de problemas, identificación de patrones y abstracción.
- Abordar diferentes problemas lógicos que puedan ser significativos para la enseñanza de conceptos de programación en la Educación Primaria y Secundaria.
- Desarrollar estrategias y heurísticas de resolución y formalización de problemas lógicos y algorítmicos.

Contenidos mínimos

- La introducción de la computadora en la escuela obligatoria: Paradigmas técnico, utilitario, integrador, y lingüístico. Programas nacionales y provinciales de alfabetización digital.
- Presentación intuitiva de algunos conceptos de programación para el modelado y la resolución de problemas: variable, estado, acción, secuencia. Habilidades de la programación: Abstracciones y representaciones.
- Revisión histórica del nacimiento de las Ciencias de la Computación. El programa de Hilbert. La paradoja de Russell. La máquina universal.
- Competencias de resolución de problemas lógicos, de ingenio, y de estrategias para desarrollar competencias relativas a búsqueda de variantes, descomposición de un problema en subproblemas, recursión, ordenamiento.
- Resolución de problemas. Tácticas, estrategias, heurísticas. Problemas combinatorios. Árboles.
- Estrategias de formalización de las soluciones de los problemas.
- Representación de la información.

Bibliografía

- Bell, T.; Alexander, J.; Freeman, I. y Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology, 13(1), 20-29. Disponible en : <http://csunplugged.org/wp-content/uploads/2014/12/unpluggedTeachersDec2008-Spanish-master-ar-12182008.pdf>
- Brennan, K. y Resnick, M. (2012). Nuevos marcos de referencia para estudiar y evaluar el desarrollo del pensamiento computacional. American Educational Research Association (AERA). Disponible en: <http://eduteka.icesi.edu.co/pdfdir/EvaluarPensamientoComputacional.pdf>
- Cotik, V. y Monteverde, H. (2016). Evolución de la enseñanza de la informática y las TIC en la Escuela Media en Argentina en los últimos 35 años. Virtualidad, Educación y Ciencia. Año 7.

Número 12 - 2016. Disponible en:

<https://revistas.unc.edu.ar/index.php/vesc/article/download/14793/14732>. Consultado: 11 de agosto de 2016.

- Levis, D. (2007). Enseñar y aprender con informática/ enseñar y aprender informática. Medios informáticos en la escuela argentina. En Cabello, R. y Levis D. S. (Comps.) (2007) Medios informáticos en la educación: a principios del siglo XXI. Buenos Aires: Prometeo. Disponible en: https://tecnoeducativas.files.wordpress.com/2010/07/u2-y-3-levis_pav06.pdf
- Valdivia, A. O. (2003). El construccionismo y sus repercusiones en el aprendizaje asistido por computadora. Contactos, 48, 61-64. Disponible en: <http://www.izt.uam.mx/newpage/contactos/anterior/n48ne/construc.pdf>
- Vicario Solórzano, C. M. (2009). Construccionismo. Referente sociotecnopedagógico para la era digital Innovación Educativa, vol. 9, núm. 47, abril-junio, pp. 45-50 Instituto Politécnico Nacional Distrito Federal, México. Disponible en: <http://www.redalyc.org/pdf/1794/179414895005.pdf>
- Zapata-Ros, M. (2015). Pensamiento computacional: una nueva alfabetización digital. Revista de Educación a Distancia, (46). Disponible en: <http://www.um.es/ead/red/46/zapata.pdf>
- Backhouse, Roland. Algorithmic Problem Solving. Wiley 2011.
- Paenza, Adrián. Matemática estás ahí?. (Y otras obras complementarias, ver <http://cms.dm.uba.ar/material/paenza/>)
- Pólya, George. Cómo Plantear y Resolver Problemas. Editorial Trillas, 1965
- Smullyan, Raymond. ¿Cómo se llama este libro? Editorial Catedra, 1981

Ensamble y Programación de un Robot

Justificación

En este módulo se trabaja sobre los conceptos básicos de programación mediante el desarrollo de un proyecto de programación de un robot. De esta manera, se proponen nuevas herramientas de enseñanza y aprendizaje de la programación.

Articulación con la práctica de la enseñanza: en este módulo se seleccionan recursos virtuales para la enseñanza de la programación de robots (tales como tutoriales para programación de robots o robots virtuales). En un informe breve se describen los criterios de selección del recurso teniendo en cuenta la edad de los alumnos a quienes va dirigido y los conceptos que se buscan enseñar, así como las potencialidades y limitaciones de enseñar programación con el recurso seleccionado.

Objetivos

- Familiarizarse con la programación de robots como recurso para enseñar en la escuela.
- Entender los mecanismos de funcionamiento de una computadora tales como un transformador de estados, a partir de la ejecución de instrucciones simples, y reforzar la relación entre hardware y software.
- Comprender cómo operan las nociones centrales de los lenguajes: formas de abstracción,

procedimientos y funciones. Constructor secuencial. Constructor de alternativa. Constructor de repetición. Variables. Asignación.

- Valorizar la importancia de la computación en tanto automatización de tareas.
- Identificar la permanencia de los conceptos centrales de la programación y su independencia de la plataforma de programación. Introducir el concepto de máquina universal.

Contenidos mínimos

- Introducción a la arquitectura de las computadoras. Definición de “robot” y de los diferentes componentes del mismo: sensores, motores, placas de control, microcontrolador, memoria, puertos, entre otros.
- Concepto de máquina universal.
- Conceptos de programación: formas de abstracción, procedimientos y funciones. Constructor secuencial. Constructor de alternativa. Constructor de repetición. Variables. Asignación.
- Conceptos de entrada y salida, a través de la programación del robot para el uso de sus sensores.
- Formatos de distribución de programas: código fuente, código binario, imágenes de disco.
- Hardware: microcontrolador, memoria volátil y no volátil, sensores, control de voltajes y tensiones, señales digitales y analógicas, puertos.
- Controladores, periférico. Conceptos básicos de arquitectura de computadora.

Bibliografía

- Gouws, L. A., Bradshaw, K., & Wentworth, P. (2013, July). Computational thinking in educational activities: an evaluation of the educational game light-bot. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education (pp. 10-15). ACM.
- Martinez, C., Gomez, M. J., & Benotti, L. (2015, June). A Comparison of Preschool and Elementary School Children Learning Computer Science Concepts through a Multilanguage Robot Programming Platform. In Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (pp. 159-164). ACM.
- Tec, B., Uc, J., Gonzalez, C., García, M., Escalante, M., & Mantañez, T. (2010). Análisis comparativo de dos formas de enseñar matemáticas básicas: robots lego nxt y animación con scratch. In Memorias de la Conferencia Conjunta Iberoamericana sobre Tecnologías para el Aprendizaje (pp. 103-109).
- Osorio, J. A. C., Salazar, O. A., & Vargas, J. A. M. (2009). El entorno lego mindstorms en la introducción a la robótica y la programación. *Scientia et Technica*, 1(41), 42-45.

Administración y Configuración de Herramientas Informáticas

Justificación

Este módulo aborda el uso del hardware, los sistemas operativos y las redes desde la perspectiva del docente como usuario crítico. Así mismo, la discusión sobre el funcionamiento de éstos permitirá reforzar el concepto de máquina universal y de capas de abstracción.

Articulación con la práctica de la enseñanza: en este módulo se elabora una propuesta de formación docente a través del uso y la instalación de hardware y software necesario para abordar los objetivos de enseñanza de la programación en el aula.

Objetivos

- Entender a la computadora y otros dispositivos programables como máquinas universales cuyo funcionamiento depende del programa y es independiente de la tecnología específica.
- Entender a los Sistemas Operativos como una capa más de abstracción que permite el ocultamiento de la tecnología específica, en este caso el hardware.
- Familiarizarse con los conceptos básicos de arquitectura de las computadoras y sistemas operativos para adquirir confianza en el entorno de trabajo.
- Conocer los conceptos básicos de administración y configuración de redes y sistemas operativos para instalar y administrar software educativo.
- Distinguir software privativo y software libre y discutir las implicaciones filosóficas y técnicas de su uso en el ámbito educativo y la vida cotidiana. Discutir sobre privacidad y soberanía de los datos.
- Distinguir hardware abierto y cerrado y discutir sobre la posibilidad de las formas de apropiación del objeto tecnológico.

Contenidos Mínimos

- Identificación de las componentes principales de una computadora. Nociones básicas del funcionamiento de una computadora moderna.
- Almacenamiento.
- Capacidad de armar/desarmar/cambiar las partes de una computadora.
- Detección y solución de problemas de una computadora en cualquiera de sus formas.
- Nociones básicas de un sistema operativo, diferenciación con la interfaz de usuario, noción de proceso vs. programa. Búsqueda de soluciones a problemas de hardware y software en internet.
- Conceptos de software libre y software privativo.
- Nociones básicas de redes de datos.
- Formatos de distribución de programas: código fuente, código binario, imágenes de disco.
- Compilación de programas distribuidos como código fuente.

Bibliografía

- Aguerre, C., Larghi, S. B., Calamari, M., Fontecoba, A., Gaztañaga, M., Moguillansky, M., ... & de León, J. P. (2010). La apropiación de las TIC por jóvenes de sectores populares urbanos en espacios de acceso público. *Revista Argentina de Estudios de Juventud*, 1(3).
- Andrew Huang, *Hacking the Xbox: An Introduction to Reverse Engineering*, 2003.
- Andrew Huang, *The Death of Moore's Law Will Spur Innovation*, *IEEE Spectrum*, 52(4), 2015.

Introducción a los lenguajes de programación a través de animaciones y videojuegos

Justificación

Este módulo se orienta, a través de proyectos específicos, a la realización de animaciones/videojuegos mediante el uso de una herramienta de enseñanza de programación. Se abordará también la potencialidad de los proyectos integrados por áreas como recurso para la enseñanza de la programación.

Articulación con la práctica de la enseñanza: en este módulo se diseña una clase de programación a través de recursos “animaciones”. Los docentes recuperan conceptos centrales de la programación y diseñan estrategias de enseñanza.

Objetivos

- Familiarizarse con la programación de animaciones y videojuegos como recursos para enseñar en la escuela.
- Promover el diseño de proyectos integradores en las escuelas que articulen con la enseñanza de la programación.
- Comprender cómo operan las nociones centrales de los lenguajes: formas de abstracción, procedimientos y funciones. Constructor secuencial. Constructor de alternativa. Constructor de repetición. Variables. Asignación.

Contenidos Mínimos

- Introducción a la programación.
- Lenguaje de Programación. Introducción al lenguaje de programación elegido (por ejemplo Python).
- Entornos didácticos para introducirse en el mundo de la programación a partir de videojuegos tales como Pilas Engine.
- Estructuras de control: formas de abstracción, procedimientos y funciones. Constructor secuencial. Constructor de alternativa. Constructor de repetición. Variables. Asignación. Noción de Objeto.
- Abstracción: diseño, modelización, y división del problema.
- La narrativa como herramienta discursiva de metacognición.
- El pensamiento computacional y sus componentes.

- El trabajo por proyectos desde la perspectiva de la interdisciplinariedad.
- Planificación de la enseñanza, el guión conjetural como género alternativo para el diseño de una secuencia didáctica. Los principios del diseño y su adaptación para la diagramación de la enseñanza de la programación mediante el desarrollo de videojuegos y animaciones.

Bibliografía

- Bahit, E. (2013). Instalación de Bazaar. En Python para principiantes. LIBROSWEB. Disponible en: https://librosweb.es/libro/python/capitulo_2.html
- Bosch, C.; Miranda, J.; Sangiorgio, M.; Acuña, I.; Michelini, Y.; Marengo, L. y Godoy, J. C. (2016). Efecto del entrenamiento con un juego de estrategia en tiempo real sobre la toma de decisiones en adolescentes. Psiencia. Revista Latinoamericana de Ciencia Psicológica, 8. Disponible en: <http://www.psiencia.org/8/1/22>
- Gee, J. (2004). Lo que nos enseñan los videojuegos sobre el aprendizaje y el alfabetismo. Málaga: Enseñanza Abierta de Andalucía - Aljibe.
- Papert, S. (1995). La máquina de los niños. Replantearse la educación en la era de los ordenadores. Buenos Aires: Paidós.
- Resnick, M. y Rosenbaum, E. (2013). Designing for Tinkerability. En Honey, M. y Kanter, D. (Eds.) Design, Make, Play: Growing the Next Generation of STEM Innovators, p. 163-181. Londres: Routledge. Disponible en: <http://web.media.mit.edu/~mres/papers/designing-for-tinkerability.pdf>
- Resnick, M. (2013). Learn to code, code to learn. EdSurge, May. Disponible en: <http://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf>

Seminario de Trabajo Final I

Justificación

En este espacio se realizará un ensayo que describa en una primera sección, el contexto nacional, provincial y local en el que se introduce la informática en la escuela. Asimismo, se analizarán las propuestas elaboradas en los módulos “Lógica, Programación y su Enseñanza”, “Ensamble y Programación de un Robot” y/o “Introducción a lenguajes de programación a través de animaciones y videojuegos” a la luz de diferentes aportes teóricos y de la práctica de la enseñanza.

Objetivos

- Ofrecer conceptos y herramientas de planeamiento educativo y didáctica que permitan a los docentes elaborar su trabajo final de postítulo.
- Generar oportunidades de aprendizaje desde la práctica docente.
- Fomentar el trabajo en equipo entre los docentes.

Contenidos Mínimos

- Planificación de la enseñanza. Criterios de selección de contenidos. Tipo de curriculum.

- Organización y secuenciación de contenidos
- Introducción a herramientas de investigación educativa. Formulación de la pregunta, identificación de dimensiones y variables. Diseño de herramientas de recolección de datos. Estrategias de análisis.

Bibliografía

- Herreras, E. B. (2004). La docencia a través de la investigación–acción. Revista Iberoamericana de Educación. Recuperado de <http://www.rieoei.org/deloslectores/682Bausela.PDF>
- García Cabrero, Benilde, Loredó Enríquez, Javier, & Carranza Peña, Guadalupe. (2008). Análisis de la práctica educativa de los docentes: pensamiento, interacción y reflexión. Revista electrónica de investigación educativa, 10(spe), 1-15. Recuperado en 12 de febrero de 2016, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1607-40412008000300006&lng=es&tlng=es
- Montero Mesa, M. L., Gewerc Barujel, A., & Jornet, J. M. (2003). El portafolios como herramienta de análisis en experiencias de formación on-line y presenciales.
- Font, V., Planas, N., & Godino, J. D. (2010). Modelo para el análisis didáctico en educación matemática. *Infancia y Aprendizaje*, 33(1), 89-105.
- Sage, S., & Torp, L. (1999). *El aprendizaje basado en problemas*. Ed. Amorrortu.

Resolución de problemas de programación a partir del diseño de juegos interactivos

Justificación

En este módulo, se propone abordar conceptos básicos de programación y estructuras de datos a partir de la resolución de problemas informáticos. Además se plantea como desafío implementar juegos de computadora que puedan ser utilizados para la enseñanza. Se propone como proyecto el diseño de un juego interactivo como el Carrera de Mente, o algún juego de ingenio que pueda servir luego como herramienta para la enseñanza en el aula.

Articulación con la práctica de la enseñanza: La apropiación de la programación, como herramienta para el diseño de nuevas aplicaciones educativas puede contribuir a que los docentes experimenten otras dimensiones de saber programar y a la vez, sentirse cómodos en el manejo conceptual de la disciplina.

Objetivos

- Fomentar estrategias de enseñanza por descubrimiento y competencias tales como resolución de problemas, identificación de patrones y abstracción.
- Afianzar los conceptos básicos de programación.
- Introducir estructuras de datos básicas como listas, tuplas y diccionarios
- Profundizar los contenidos en técnicas de resolución de problemas computacionales.
- Debatir sobre problemas concretos en la escuela o la vida social que pueden ser resueltos con un programa computacional.

Contenidos Mínimos

- Concepto de algoritmo.
- Programación orientada a objetos y tipos abstractos de datos.
- Estructuras de datos (Tuplas, Listas, diccionarios)
- Tipos de archivos para representación de información.
- Funciones. Encapsulamiento. Tareas repetitivas. Generalización por abstracción.
- Concepto de tipo de datos
- Resolución de problemas. Tácticas, estrategias, heurísticas. Problemas combinatorios. Árboles.
- Estrategias de formalización de la soluciones de los problemas.
- Diferencia entre editor, intérprete, compilador.
- Nociones básicas sobre compiladores/intérpretes.
- Noción de Software Libre

Bibliografía

- Bell, T.; Alexander, J.; Freeman, I. y Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology, 13(1), 20-29. Disponible en : <http://csunplugged.org/wp-content/uploads/2014/12/unpluggedTeachersDec2008-Spanish-master-ar-12182008.pdf>
- Backhouse, Roland. Algorithmic Problem Solving. Wiley 2011.

Introducción a la programación orientada a objetos a partir del desarrollo de un sistema de votación

Justificación

Este módulo tiene por objetivo brindar una práctica de la programación más profunda que la adquirida en módulos anteriores, utilizando un lenguaje de programación más avanzado, e introduciendo conceptos sofisticados de definición de abstracciones de datos, de reuso, de modularización y descomposición de programas, específicamente vinculados a la Orientación a

Objetos. Por otra parte, en este módulo se introducirán cuestiones más avanzadas, vinculadas a la Ingeniería de Software, y particularmente sobre aspectos fundamentales en la calidad de sistemas de software actuales, como la confidencialidad de la información y la seguridad, entre otros. Estos contenidos serán brindados a través del desarrollo de un sistema de votación simple y de baja escala, que permitirá destacar además la importancia de propiedades no funcionales de sistemas de software, como la confidencialidad y seguridad, fomentando una mirada crítica de la tecnología.

Articulación con la práctica de la enseñanza: en este módulo se revisa la unidad didáctica diseñada en el cursado del “Seminario de Trabajo Final I”, buscando incorporar nuevos conceptos de las ciencias de la computación.

Objetivos

- Afianzar conceptos básicos de programación.
- Introducir conceptos de modularidad, reuso y descomposición de programas a través de la programación orientada a objetos.
- Introducir el concepto de abstracción de datos, nuevas estructuras de datos, y afianzar el uso de las ya conocidas.
- Implementar programas de complejidad acotada, en un lenguaje estándar de programación.
- Introducir conceptos más avanzados de sistemas de software actuales, como la importancia de aspectos no funcionales de sistemas de software, en particular la seguridad informática y la confidencialidad de la información.
- Discutir sobre propiedad intelectual y licencias. Analizar las implicancias de éstas en la apropiación del conocimiento y en el modo de construcción de software que habilitan.
- Analizar el uso de licencias libres aplicadas a la producción artística.
- Construir una mirada de la computación en tanto ciencia de la abstracción, y entender cómo se construye un programa para poder elaborar otros sobre desarrollos existentes.

Contenidos Mínimos

- Conceptos de programación imperativa. Variables. Asignación. Constructor secuencial. Constructor de alternativa. Constructor de repetición. Formas de abstracción procedimental, procedimientos y funciones.
- Diferencia entre editor, intérprete, compilador.
- Tipos de datos básicos y estructurados.
- Conceptos básicos de programación orientada a objetos. Clases, objetos. Ocultamiento de la información. Concepto de abstracción de datos.
- Representación de la información. Tipos abstractos de datos. Estructuras de datos.
- Estructuras de datos básicas.
- El Proceso de Desarrollo de Software. Etapas fundamentales. Análisis, diseño, implementación, validación y verificación. Procesos ágiles de desarrollo. Diseño orientado a objetos. Pruebas de software.

- Software libre y privativo.
- Propiedades no funcionales de sistemas de software. Principios de encriptación. Principios de seguridad y confidencialidad de la información.
- Conceptos de filosofía de la técnica.

Bibliografía

- A. Downey, *Think Python*, 1st Edition, O'Reilly, 2012.
- R. González Duque, *Python para Todos*, dist. bajo Creative Commons, 2010.
- D. Gries y P. Gries, *Multimedia Introduction to Programming using Java*, Springer, 2005.
- Kenkeiras, *De 0 a Python*, dist. bajo Creative Commons, 2011.
- B. Liskov, *Program Development in Java: Abstraction, Specification and Object-Oriented Design*, Addison-Wesley, 2000.
- B. Meyer, *Construcción de Software Orientado a Objetos*, Prentice-Hall, 1998.
- B. Meyer, *Touch of Class: Learning to Program Well with Objects and Contracts*, Springer, 2009
- G. Polya, *How to Solve it*, 2nd Edition, Princeton University Press, 1957.
- D. Parnas, *A Technique for Software Specification with Examples*, *Communications of the ACM* 15(12), ACM, 1972.
- M. Pilgrim, *Dive into Python*, Springer, 2004.
- Z. Shaw, *Learn Python the Hard Way, A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code (3rd Edition)*, Addison-Wesley, 2013.

Procesamiento de datos a partir del análisis de bases de datos públicas

Justificación

Se realiza un proyecto de ciencia de datos con un programa simple. Los docentes llevarán a cabo las principales tareas de este tipo de proyectos: recolección, análisis, procesamiento y visualización de datos a partir de una librería de datos pública. La experiencia requiere la apropiación de conceptos centrales de procesamiento de datos (adquisición y visualización de datos) y su articulación con otros temas de programación, como minería de datos. Asimismo, se analizarán conceptualmente nuevas tecnologías en términos de sus licencias y mecanismos de recolección, almacenamiento y análisis de datos personales (ej.: cámaras de seguridad, manipulación de información de contactos o actividades, etc.) y las bases filosóficas de la protección legal de los derechos a la privacidad.

Articulación con la práctica de la enseñanza: en este módulo, a partir de la experiencia de minería de datos, se realiza una propuesta para analizar los datos recolectados sobre la experiencia de enseñanza de programación planteada en el módulo "Seminario de Trabajo Final I".

Objetivos

- Afianzar los conceptos y habilidades de programación aprendidos en los módulos anteriores.

Programación orientada a objetos, estructuras de datos básicas, representación de la información.

- Familiarizarse con conceptos centrales de la recolección, procesamiento y análisis de datos.
- Desarrollar un programa sencillo utilizando bibliotecas predefinidas de un lenguaje de programación como Python.
- Discutir sobre las bases filosóficas de la protección legal de los derechos a la privacidad entendiendo el funcionamiento de ciertos servicios de internet y las posibilidades de la minería de datos y el procesamiento de la información.

Contenidos Mínimos

- Programación orientada a objetos.
- Tratamiento de estructuras de datos básicas (listas y registros) desde un punto de vista de alto nivel (recorridos, búsquedas, totalizaciones).
- Funciones. Encapsulamiento. Tareas repetitivas. Generalización por abstracción.
- Bibliotecas de los lenguajes de programación. Acceso y usos. Programación como experimentación.
- Procesamiento lineal de datos.
- Nociones de funcionamiento de programas heurísticos y de inteligencia artificial, como ser asistentes de texto predictivo, clasificadores, etc.
- Nociones de seguridad y minería de datos.
- Privacidad de los datos. Servicios de internet. Almacenamiento en la nube.
- Metadatos.
- Programación Secuencial: stream programming.
- Diferencia entre editor, intérprete, compilador.
- Adquisición de datos (procesamiento de strings, formatos de archivo, reconocimiento de caracteres: ocr).
- Tipos de archivos: imágenes, texto, etc.

Bibliografía

- [Adèr, H.J.](#) (2008). Chapter 14: Phases and initial steps in data analysis. In H.J. Adèr & G.J. Mellenbergh (Eds.) (with contributions by D.J. Hand), *Advising on Research Methods: A consultant's companion* (pp. 333–356). Huizen, the Netherlands: Johannes van Kessel Publishing.
- [Adèr, H.J.](#) (2008). Chapter 15: The main analysis phase. In H.J. Adèr & G.J. Mellenbergh (Eds.) (with contributions by D.J. Hand), *Advising on Research Methods: A consultant's companion* (pp. 333–356). Huizen, the Netherlands: Johannes van Kessel Publishing.
- Tabachnick, B.G. & Fidell, L.S. (2007). Chapter 4: Cleaning up your act. Screening data prior to analysis. In B.G. Tabachnick & L.S. Fidell (Eds.), *Using Multivariate Statistics, Fifth Edition* (pp. 60–116). Boston: Pearson Education, Inc. / Allyn and Bacon.

Taller de Desarrollo de Aplicaciones Web

Justificación

Muchas aplicaciones modernas son aplicaciones en red. En este taller se introducen los conceptos de redes, telecomunicaciones con un enfoque de arriba hacia abajo, es decir, desde una aplicación, pasando por las capas inferiores de una suite de protocolos, en particular, TCP/IP.

En este módulo se desarrollará una simple aplicación HTML5 interactiva educativa. Se abordarán temas tales como introducción a las redes y las telecomunicaciones, protocolos de comunicación, aplicaciones y la web.

Articulación con la práctica de la enseñanza: en este módulo se realiza el diseño de un sistema web con contenido educativo que incluya animaciones y elementos interactivos, usando canvas y svg. Se utilizarán herramientas de software libre para gráficos vectoriales, capturas de pantalla y videos, y juegos como herramientas didácticas. Introduciremos los aspectos de accesibilidad para usuarios con limitaciones visuales y/o auditivas. El producto obtenido podrá utilizarse en modo solo cliente o modo cliente servidor.0 Para este último modo se proveerá y analizará un pequeño servidor de aplicaciones.

Objetivos

- Afianzar los conceptos y habilidades de programación aprendidos en los módulos anteriores.
- Introducir a los conceptos de redes, telecomunicaciones y desarrollo de aplicaciones en red.
- Comprender el funcionamiento de las aplicaciones en red, específicamente las aplicaciones web.
- Aproximar a las tecnologías web modernas y al desarrollo de aplicaciones interactivas, específicamente, materiales educativos interactivos.
- Desarrollar y usar documentos web como material educativo interactivo.
- Introducirse a las técnicas de animación de gráficos svg usando el canvas.
- Experimentar el uso de un navegador web como plataforma de desarrollo de aplicaciones.
- Desarrollar aplicaciones web educativas multiplataforma.
- Análisis y discusión sobre la interactividad como herramienta didáctica.

Contenidos Mínimos

- Conceptos de programación distribuida (mensajes, sincronización, etc).
- Conceptos básicos de redes: ruteo, capas (física, transporte, etc.). TCP/IP. DNS.
- Modelo cliente-servidor
- Aplicaciones web.
- Concepto de interfaz. Interfaces web.
- Introducción a técnicas de animación de gráficos svg usando el canvas.
- Representación de datos. Tipos de archivos.

- Nociones de servicios en Internet. Ej. Almacenamiento en la nube.
- Introducción a aspectos de accesibilidad.

Bibliografía

- CONTENT PRODUCTION FOR E-LEARNING ENVIRONMENTS WITH HTML5. C. Guller, S. Gumus, E Guller. 4º International Conference on Education and New Learning Technologies. Barcelona, España. 2012.
- "El Gran Libro de HTML5, CSS3 y Javascript". Juan Diego Gauchat. Ed. Marcombo. ISBN: 78-84-267-1782-5.
- Dive Into HTML5. Mak Pilgrim. <https://github.com/mislav/diveintohtml5>. Free e-book. Licence: Creative Commons.
- Computer networking. A top-down Approach. Kurose, Ross. Pearson Education Inc. ISBN-10: 0-13-285620-4. 2013.

Seminario de Trabajo Final II

Justificación

En este módulo se desarrolla a lo largo del segundo año de la especialización. Los docentes deben diseñar y poner en práctica un proyecto de enseñanza de la programación enmarcado en el análisis del Seminario de Trabajo Final I, con los aportes de los módulos posteriores. Para esta etapa se contempla 20hs de trabajo en la escuela y 10 horas de tutorías. Durante las 10 horas restantes de este módulo se analizarán los resultados de la implementación realizada. El foco de este proyecto será la reflexión sobre la puesta en práctica de las clases de programación elaboradas por los docentes, recuperando para ello las producciones que realizaron los alumnos y los datos que recolectaron sobre la experiencia. Para lograr una reflexión sistemática y significativa se abordarán nociones vinculadas a la investigación, tales como definición del problema y estrategias de análisis. En un informe final los docentes presentarán el análisis de resultados de las propuestas implementadas y reflexionarán sobre implicancias y derivaciones posibles de la experiencia. Para elaborar el informe final, se espera que los docentes revisen los materiales teóricos y prácticos ofrecidos durante todo el postítulo.

Objetivos

- Desarrollar estrategias de reflexión sistemática sobre la práctica de la enseñanza.
- Apropiarse de competencias propias de la investigación educativa.
- Reflexionar sobre la enseñanza de la programación en computación.
- Elaborar y presentar el trabajo final de carrera.

Contenidos mínimos

- Herramientas de investigación educativa. Formulación de la pregunta, identificación de dimensiones y variables. Diseño de herramientas de recolección de datos. Estrategias de análisis.
- Análisis de la clase. Los segmentos de una clase. Coherencia de la propuesta.
- Escritura de informes. Representación de datos.

Bibliografía

- Berrocoso, J. V., Sánchez, M. R. F., & Arroyo, M. D. C. G. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, (46).
- Olabe, X. B., Basogain, M. Á. O., & Basogain, J. C. O. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. *Revista de Educación a Distancia*, (46).
- Guías para escribir informes de investigación educativa.
- Redes de computadoras. 4ta edición. Andrew S. Tanenbaum. Pearson, Prentice Hall. ISBN: 970-26-0162-2.

Evaluación

Para recibir el título de Especialista, los cursantes deberán aprobar la totalidad de los módulos de este plan de estudios así como el módulo final en el que se elaborará el diseño de un proyecto de enseñanza de la programación (selección del tema, contenidos, integración en el currículum, objetivos y actividades) que se pondrá en práctica obligatoriamente. Podrán recibir un certificado de actualización aquellos docentes que aprueben todos los módulos del primer año y el trabajo final del último módulo en que se analizarán las propuestas elaboradas en los módulos “Lógica, Programación y su Enseñanza”, “Ensamble y Programación de un Robot” y/o “Introducción a lenguajes de programación a través de animaciones y videojuegos” recuperando aportes de experiencias de enseñanza de programación y enmarcándolas en un contexto más general de incorporación de tecnología en las aulas.

El resultado de la evaluación de cada módulo se consignará con los términos: APROBADO (con calificación igual o superior a seis) y DESAPROBADO (con calificación inferior a seis).

Modalidad de Evaluación: Sosteniendo coherencia con la propuesta de enseñanza por proyectos, cada módulo requerirá que los docentes cursantes elaboren un proyecto integrador. La instancia de evaluación final será la presentación de los proyectos a docentes y colegas en una exposición oral y acompañada de un informe.