|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Taller de Programación II | | | | **4.3.029** |
| **Departamento al que pertenece** | | | **Director** | | |
| **Comunicación y Publicidad** | | | Dis. Fernanda Kuroda | | |
| **Carga** **horaria** | | | **Fecha de aprobación en el**  **Consejo de Facultad y N° de Acta** | | |
| 68 hs | | | 28/07/2020 – Acta n° 309 | | |
| Carrera(s) en la que se dicta | | | | Código(s) **Carrera(s)** | |
| Licenciatura en Comunicación Digital e Interactiva | | | | 751 | |
| Código(s) Correlativa(s) Precedente(s) | | Código(s) Correlativa(s) Subsiguiente(s) | | Código(s) Carrera(s) | |
|  | |  | |  | |
| Firmas | | | | | |
| Aprobación del Director de Departamentoemisor | | | **Aprobación Decano(s)** | | |
|  | | |  | | |

### I – Fundamentación

La asignatura Taller de Programación II es una materia teórico-práctica del segundo año de la Licenciatura en Comunicación Digital e Interactiva, que se propone transmitir y afianzar los conceptos básicos de programación vistos en Taller de Programación I, como una herramienta necesaria para la comunicación digital.

El futuro comunicador, además de poseer conocimientos en esta disciplina, contará con la posibilidad de participar en equipos multidisciplinarios con programadores, en donde deberá utilizar un léxico específico que esta asignatura aportará. En tal sentido, la materia brindará las nociones lógicas existentes detrás de los sistemas informáticos que son necesarias para que el alumno sepa sobre la construcción y posibilidades que brinda la realización de un software de forma personal, y que le permita enfrentarse con seguridad a la interacción con profesionales informáticos.

Se favorecerá que el estudiante sea capaz de reconocer, describir y analizar distintos tipos de construcciones lógicas del software a través del conocimiento de estructuras y programas informáticos básicos que consideren buenas prácticas en el desarrollo de software.

### II – Objetivos de aprendizaje (a) y de enseñanza (b)

1. **Se espera que los alumnos logren:**

* Desarrollar competencias lógicas, analítica y de programación que se integren al perfil profesional como a los objetivos de la carrera, y que les permitan desenvolverse adecuadamente en las distintas situaciones del software.
* Adquirir capacidades básicas de análisis, elaboración y corrección de sus propios programas informáticos.
* Conocer recursos para superar las propias dificultades con la programación y los errores que ellas conllevan.
* Ampliar la riqueza tecnológica, haciendo uso de buenas prácticas en el desarrollo de software.
* Desarrollar capacidades de investigación y visión crítica de una solución de software.

### Se espera que el docente del curso logre:

* Promover la práctica analítica y funcional sobre la programación, al servicio de una mayor comprensión lógica y una utilización de buenas prácticas.
* Contribuir a la formación de profesionales con una sólida base formal que les permita enfrentarse con seguridad a las nociones básicas en lógica de programación.
* Proporcionar herramientas estructurales de metodología para la creación de programas informáticos, junto con consejos y recursos tecnológicos para programar de la manera más eficiente.
* Promover el pensamiento crítico sobre una solución de software.

**III- Unidades temáticas**

**Unidad I: Revisión de conceptos previos**

El lenguaje Python. Sintaxis y semántica. Variables y tipos de datos. Expresiones y operadores. Estructuras de control condicionales. Resolución de problemas sencillos a través de algoritmos.

**Unidad II: Entorno de ejecución (IDE)**

Introducción a un entorno de desarrollo integrado. Elementos que componen un entorno de desarrollo integrado. Elección de un entorno de desarrollo integrado. Instalación de un entorno de desarrollo integrado. Aplicación de un programa en Python en un entorno de desarrollo integrado.

**Unidad III: Estructuras iterativas**

Introducción a listas, json y diccionarios. Estructura general de una lista, json y diccionario. Aplicación de listas, json y diccionarios en un programa en Python.

**Unidad IV: Funciones predefinidas**

Introducción a las funciones predefinidas (built-in functions) en Python. Aplicación de funciones predefinidas (built-in functions) en un programa en Python.

**Unidad V: Funciones**

Introducción a las funciones en Python. Elementos que componen a una función. Pasaje de parámetros. Alcance de variables. Aplicación de funciones en un programa en Python.

**Unidad VI: Archivos (Texto)**

Introducción a archivos en Python. Lectura y escritura de un archivo en Python. Aplicación de funciones predefinidas en archivo. Aplicación de un archivo (texto) en un programa en Python.

**Unidad VII:** **PIP (Package Installer for Python)**

Introducción a PIP. Aplicación de PIP en un programa en Python. Elementos que componen a un paquete en Python. Aplicación de un paquete en un programa en Python.

**Unidad VII: Github**

Introducción a Github. Elementos que componen a Github. Aplicación de Github en un programa en Python.

**Unidad VIII: Recursos para trabajo final**

Investigación para posibles temas de trabajo final. Consideraciones y formas de presentación. Tipos de trabajo final. Aplicación de referencia de un trabajo final modelo.

**IV- Estrategias de enseñanza**

Se trabajará bajo la modalidad de taller en gabinetes informáticos y aula convencional, con exposiciones para el desarrollo de ideas teórico-prácticas que sirvan de marco y fundamento a la introducción a la programación.

Las consignas se orientarán a la comprensión y el desarrollo de capacidades en programación informática, entendida como una herramienta necesaria para la comunicación digital.

Se desarrollarán trabajos de análisis y desarrollo de programación, para comprender y valorar la capacidad de lógica utilizada con el objetivo de aplicar estrategias en la elaboración de programas por parte del alumno.

**V- Medios y recursos**

* Aulas informáticas (UADE *Labs*).
* Proyector y PC.
* Material audiovisual.

**VI- Modalidad de Evaluación**

# VI.I         Condiciones de aprobación de la cursada

* Cumplimiento del 75 % de asistencia y de los horarios de ingreso y salida fijados por la universidad para cada turno.
* Aprobación de la/s evaluación/es parcial/es especificadas en el Cronograma de la materia.
* Aprobación del/los trabajo/s práctico/s.
* En todos los casos, la modalidad será oral o escrita, según corresponda al tipo de evaluación, y la aprobación será con una calificación igual o mayor a 4.
* Quienes resulten desaprobados o ausentes en sólo una de las evaluaciones parciales, podrán rendir un único examen recuperatorio, en la modalidad oral o escrita, según corresponda al tipo de evaluación.

# VI.II       Condiciones de aprobación de la asignatura

En el caso de materias con trabajo final, los alumnos que obtuviesen una calificación mínima de 7 (siete) puntos en cada una de las entregas parciales (al menos 1 entrega parcial), obtendrán la aprobación de la asignatura mediante la presentación y aprobación del trabajo final. Este sistema de promoción no permite rendir examen recuperatorio alguno.

Los alumnos que, habiendo aprobado las entregas parciales con un mínimo de 4 (cuatro) puntos y que no logren la calificación necesaria para la presentación del trabajo final o desaprueben esta instancia de evaluación, podrán rendir examen final de la asignatura en los 11 (once) turnos de exámenes finales consecutivos posteriores a la aprobación de la cursada.

**VII- Bibliografía de consulta**

**Básica**

* JOYANES AGUILAR, L. (2008). *Fundamentos de Programación. Algoritmos, estructura de datos y objetos.* Madrid: Mc.Graw-Hill. ISBN 978-84-481-6111-8. EAN 9788448161118.
* *The Python Language Reference*. [consulta: 9/07/20]. Español: https://docs.python.org/3/reference/

**De consulta**

* PILGRIN, M. *Inmersión en Python 3* [en línea] Traducido por José Miguel González Aguilera. Disponible gratuitamente (bajo licencia Creative Commons 3.0). [consultas: 9/07/20]: http://www.jmgaguilera.com/libro/python/traducci%C3%B3n/latex/2016/08/19/inmersion-python.html
* MARZAL VARÓ., GRACIA LUENGO, I. y GARCÍA SEVILLA, P, A*.: Introducción a la programación con Python 3* [En línea] Disponible gratuitamente bajo licencia Creative Commons. [consultas: 9/07/20]: https://openlibra.com/es/book/introduccion-a-la-programacion-con-python-3
* Documentos varios en Python.org.ar [consultas: 9/07/20]. Inglés / Español: http://www.python.org.ar/aprendiendo-python/

**VIII – Cronograma-tipo**

|  |  |
| --- | --- |
| **Clase** | **Contenidos/Actividades** |
| 1 | Presentación de la materia.  Repaso de taller de programación I (estructura iterativa, secuencial, condicional, estructuras de datos, funciones). |
| 2 | Estructuras de datos.  Ejercitación. |
| 3 | Funciones y ordenamiento de listas.  Ejercitación. |
| 4 | Github. |
| 5 | IDEs (Entorno Integrado de Desarrollo).  Bibliotecas externas y PIP (Package Installer for Python).  Ejercitación con mapas y gráficos. |
| 6 | Ejercitación para el parcial: Estructura iterativa, secuencial, condicional. |
| 7 | Ejercitación para el parcial: Estructura iterativa, secuencial, condicional. |
| 8 | Ejercitación para el parcial: Estructura de datos (listas) y funciones. |
| 9 | **PRIMER PARCIAL** |
| 10 | **DEVOLUCIÓN DE PRIMER PARCIAL**  Presentación del Trabajo Práctico y elección de tema. |
| 11 | Recursos para trabajo final: Repaso de pandas, matplotlib y otras bibliotecas útiles para utilizar en el TP. |
| 12 | Recursos para trabajo final: La importancia del storytelling y cómo presentar la información al usuario. Dashboards. |
| 13 | Recursos para trabajo final: Trabajo personalizado con cada grupo de acuerdo a las necesidades particulares de cada TP. |
| 14 | **ENTREGA Y PRESENTACIÓN DE TRABAJOS PRÁCTICOS.** |
| 15 | **ENTREGA Y PRESENTACIÓN DE TRABAJOS PRÁCTICOS.** |
| 16 | **RECUPERATORIOS / FINAL ADELANTADO** |