

## PROGRAMA ACADÉMICO

<b>Módulo</b>	Introducción a la programación con Ruby
<b>Descripción</b>	<p>El módulo introductorio de programación con Ruby entrega las herramientas y conocimientos básicos para construir scripts que puedan leer datos desde archivos y APIs, limpiar datos de acuerdo a las necesidades del negocio y generar archivos finales fáciles de procesar.</p> <p>El alumno con estos conocimientos podrá optar por profundizar estos aprendizajes o aprender a trabajar con frameworks para la construcción de aplicaciones web.</p>
<b>Competencias Generales</b>	<ul style="list-style-type: none"><li>• Instalar un entorno de trabajo para Ruby.</li><li>• Conocer las reglas sintácticas de Ruby.</li><li>• Conocer el flujo de trabajo para procesar datos desde un archivo y crear un nuevo archivo con los datos modificados.</li><li>• Construir scripts reutilizables en Ruby para procesar datos.</li><li>• Realizar conexión a APIs para la descarga o inserción de datos</li><li>• Realizar las conexiones vía HTTPS para impedir el robo de credenciales.</li></ul>

## ALCANCE

Unidad	Descripción	Competencias
Flujo	Principios básicos sobre la programación con Ruby como lenguaje, así como la introducción en los conceptos básicos de control de flujo y operadores lógicos.	<ul style="list-style-type: none"> <li>• Instalar herramientas: Editor, Ruby.</li> <li>• Ejecutar Ruby desde el terminal.</li> <li>• Ejecutar Ruby desde el editor de texto.</li> <li>• Realizar diagramas de flujo y pseudocódigo.</li> <li>• Construir aplicaciones tipo calculadora.</li> <li>• Manejar flujos y operadores lógicos.</li> <li>• Leer y transcribir diagramas de flujo con interacciones a código Ruby.</li> <li>• Analizar e implementar diagramas de flujo con repeticiones.</li> <li>• Analizar condiciones de borde.</li> <li>• Simplificar problemas con condiciones anidadas en Ruby.</li> <li>• Entender las reglas básicas de sintaxis a través del análisis léxico.</li> </ul>
Ciclos y métodos	Implementar ciclos a través de sus diferentes formas en Ruby para resolver problemas de iteración como ciclos y sumatorias. Utilizar métodos para reutilizar códigos, así como entender el alcance de nombre de las variables dentro de estos.	<ul style="list-style-type: none"> <li>• Conocer los ciclos y sus posibles aplicaciones en la programación.</li> <li>• Leer y transcribir diagramas de flujo con iteraciones a código Ruby.</li> <li>• Contar la cantidad de veces que un programa está dentro de un ciclo.</li> <li>• Realizar programas donde el usuario ingrese múltiples datos hasta que decida detenerse.</li> <li>• Ciclos y sumatorias.</li> <li>• Escribir en ruby el código de una sumatoria.</li> <li>• Diferenciar contadores de acumuladores.</li> <li>• Utilizar bloques.</li> <li>• Reconocer patrones de repetición en un</li> </ul>

		<p>ciclo.</p> <ul style="list-style-type: none"> <li>• Utilizar ciclos anidados para resolver problemas.</li> <li>• Crear métodos.</li> <li>• Parametrizar métodos.</li> <li>• Crear métodos con parámetros opcionales.</li> <li>• Crear métodos con retorno.</li> <li>• Conocer los retornos implícitos.</li> <li>• Conocer los tipos de variable.</li> <li>• Conocer el concepto de alcance.</li> <li>• Diferenciar variables locales de variables globales.</li> </ul>
Arreglos y archivos	Utilizar arreglos para almacenar y recuperar datos, realizar operaciones de reducción e iteración sobre arreglos y persistir datos utilizando archivos.	<ul style="list-style-type: none"> <li>• Manejar e iterar arreglos.</li> <li>• Leer, agregar, modificar y borrar elementos.</li> <li>• Utilizar arreglos para leer archivos.</li> <li>• Guardar datos en archivos.</li> <li>• Filtrar elementos de un arreglo.</li> <li>• Transformar elementos de un arreglo.</li> <li>• Reducir un arreglo a un solo elemento.</li> <li>• Operaciones funcionales sobre los arrays.</li> <li>• Leer la documentación del método enumerable.</li> <li>• Aplicar métodos específicos en lugar de .each a cada situación.</li> <li>• Realizar operaciones típicas de conjunto sobre arrays.</li> </ul>
Hashes y APIs	Utilizar la estructura de datos de diccionarios para resolver problemas e identificar situaciones donde es mejor utilizar diccionarios en vez de	<ul style="list-style-type: none"> <li>• Entender el objetivo de una API.</li> <li>• Conocer los principales componentes de una API.</li> <li>• Conocer la lógica de consumo de API.</li> <li>• Utilizar Postman para realizar requests a una API.</li> </ul>

	<p>arreglos.</p> <p>Interactuar con las API's, una de las principales formas en las cuales las aplicaciones pueden comunicar resultados y recibir consultas desde usuarios u otra aplicación.</p>	<ul style="list-style-type: none"><li>• Conocer los endpoints.</li><li>• Conocer y entender los verbos REST.</li><li>• Conocer el formato de archivos JSON.</li><li>• Utilizar Ruby para realizar un request a una API.</li><li>• Consumir los datos desde Ruby.</li><li>• Guardar los requests de una API en un JSON y procesarlos dentro de Ruby.</li><li>• Conocer y comprender la importancia del protocolo HTTPS.</li><li>• Utilizar Ruby para realizar un request mediante HTTPS.</li><li>• Almacenar datos en arreglos y/o hashes y procesarlos utilizando las técnicas aprendidas.</li></ul>
--	---	--

## REQUERIMIENTOS

<b>Recursos de Información**</b>	<b>Características notebook*</b> <ul style="list-style-type: none"><li>• Sistema Operativo: Mac o Linux (Para los estudiantes que posean Windows, deben realizar una partición de disco o bien cambiar el sistema operativo a Linux)</li><li>• Procesador: Procesador Intel Core i3 de 5ta generación</li><li>• RAM: 8GB, 500GB HD, 128 Disco SSD.</li></ul> <b>Plataformas y Software</b> <ul style="list-style-type: none"><li>• LMS</li><li>• ATOM (<a href="https://atom.io/">https://atom.io/</a>)</li><li>• Git (<a href="https://git-scm.com/downloads">https://git-scm.com/downloads</a>)</li><li>• GitHub / GitHub pages (<a href="https://github.com">https://github.com</a>)</li><li>• RVM (<a href="https://rvm.io/">https://rvm.io/</a>)</li><li>• Ruby 2.5.3 (<a href="https://ruby-doc.org/core-2.5.3/">https://ruby-doc.org/core-2.5.3/</a>)</li></ul>
----------------------------------	--

*\* El notebook es por cuenta de todos los participantes: docente, ayudante y alumnos.*

*\*\* Programas open source, por lo que el estudiante no necesita incurrir en gastos de licencias.*