



CAMPUS LAND

PROGRAMA DE FORMACION
DESARROLLO FULLSTACK

RUTA FORMATIVA 2

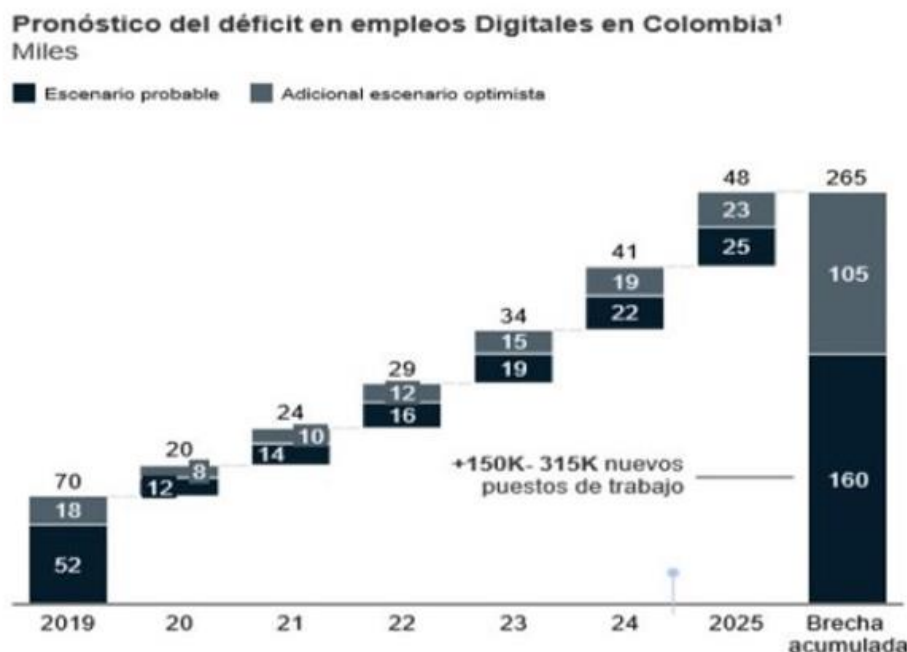
INTRODUCCION

Ante la creciente demanda de profesionales específicamente de diseño y desarrollo de software que requiere la cuarta revolución industrial, surge la iniciativa de formar el talento humano necesario para generar las habilidades digitales en ciudadanos de manera que contribuya en el mejoramiento el pensamiento computacional y el emprendimiento ante las exigencias de un mundo cada vez más competitivo y en coherencia con los retos de la industria de Tecnologías de Información (TI), sectores estratégicos y productivos.

A nivel mundial se ha detectado una importante brecha entre la necesidad de talento digital y la oferta de personal capacitado. Por ejemplo, un análisis realizado por (McKinsey & Company, 2017) con información del Observatorio TI, Informe Talent Gap realizado en Colombia por MINTIC, EAFIT & Infosys en 2017 indica que: “Colombia tiene una brecha de talento digital que llegaría a 160k-265k trabajos en 2025, donde 68k - 112k corresponderían a desarrolladores de software”. La situación se torna similar a nivel mundial y es por ello que se deben crear estrategias de formación, en especial, de programadores y desarrolladores de software.

Figura 1.

Pronóstico del déficit en empleos digitales en Colombia



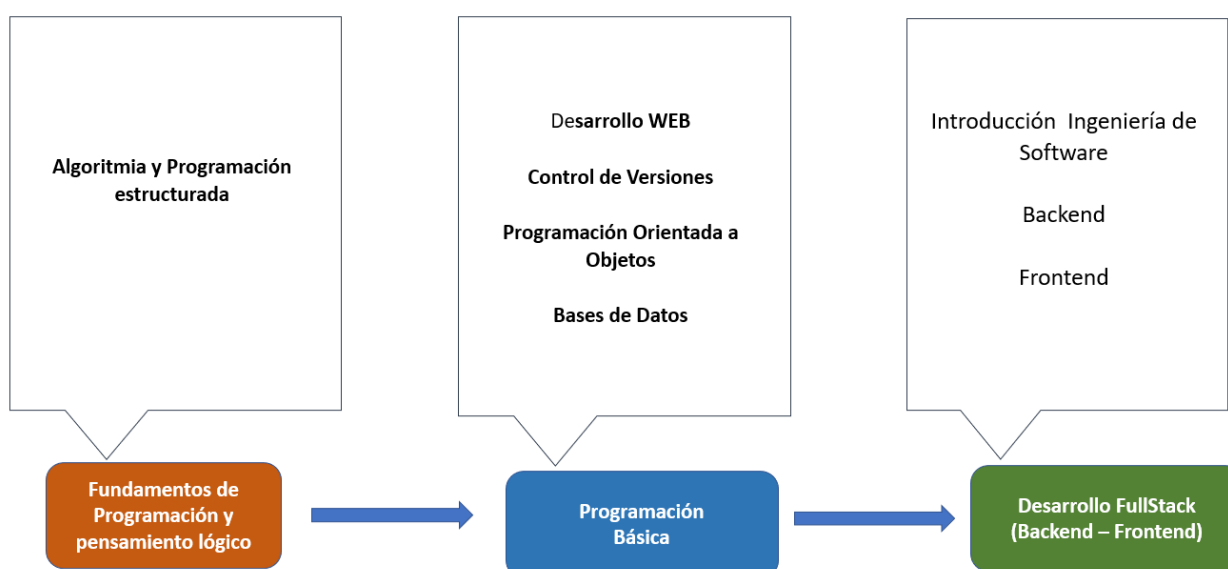
Fuente: McKinsey & Company, con información del Observatorio TI, 2017

RUTA FORMATIVA PROGRAMA DESARROLLO FULLSTACK

El programa que ofrece Campus Land sobre formación en desarrollo FullStack, en su primera iteración, ofrece la siguiente estructura de ciclos y contenidos.

Figura 2.

Estructura de la ruta formativa



Fuente: Campus Land.

Es importante mencionar que el programa desarrollo FullStack ofrecido por Campus Land, pretende la formación integral de los participantes y por ello ofrece tres aspectos en su proceso de aprendizaje:

- Desarrollo de competencias digitales, en los aspectos de programación y desarrollo de software
- Desarrollo de las competencias en una segunda lengua, para este caso inglés.
- Desarrollo de competencias en habilidades blandas, para la formación del SER.

Además, se presentan las tecnologías que incluirá cada uno de los ciclos y temas, según los requerimientos de las empresas de desarrollo de software, patrocinadores del proyecto Campus Land.

Figura 3.
Tecnologías involucradas en la ruta formativa programa desarrollo FullStack



Fuente: Campus Land.

Para el cumplimiento de esta ruta formativa, el cronograma dispuesto para su desarrollo se presenta a continuación:

[illegible]



Finalmente, la distribución de horas para cada uno de los aspectos del programa Desarrollo FullStack es el siguiente:

Aspecto	Horas
Competencias digitales en desarrollo de software y programación	640
Acompañamiento con tutores (Review)	320
Competencias en una segunda lengua (inglés)	160
Desarrollo de habilidades blandas	160
TOTAL	1280



METODOLOGIA

Con respecto a las estrategias metodológicas que se utilizarán, se mencionan:

Encuentros presenciales: Para la presentación, explicación, desarrollo de los temas y las actividades con el formador en lo concerniente con los aspectos temáticos en las áreas de programación y desarrollo de software.

Aprendizaje Basado en Retos (ABR): Como una de las principales estrategias didácticas, en este programa se implementa el Aprendizaje Basado en Retos (ABR), esta estrategia didáctica se desarrolla mediante actividades experienciales, se caracteriza por el rol protagónico del estudiante y fomenta la autorregulación y la autonomía. Hung, S. (Ed), (2015:366).

Durante el desarrollo del módulo, el estudiante se enfrentará a una serie de retos sobre problemas de manejo de información, que serán propuestos por el formador responsable. Cada reto, le ayudará a los participantes a reforzar sus conocimientos y habilidades en los diferentes ejes temáticos desarrollados en las sesiones presenciales. Además, los retos deberán ser resueltos de manera individual por cada uno de los estudiantes en los tiempos definidos por el formador, con el apoyo a nivel de asesoría por parte de los tutores del módulo.

Aprendizaje basado en proyectos (ABP): Durante el desarrollo de todo el programa de formación, los estudiantes realizarán un proyecto, el cual implica un desarrollo de software, que permita aplicar y validar el proceso de aprendizaje, El proyecto de software a desarrollar tendrá las siguientes características:

- El proyecto debe tener un contexto real.
- El proyecto debe estar estructurado para ser desarrollado durante todo el proceso de formación, con requerimientos claros para cada una de los módulos.
- El proyecto debe estar planteado para ser desarrollado de manera colaborativa por un equipo de estudiantes.
- El proyecto debe especificar los entregables en cada uno de los módulos del programa de formación junto con el conjunto de pruebas unitarias para validar su cumplimiento.

Finalmente, se presenta un cuadro comparativo características de las metodologías de Aprendizaje Basado en Retos (ABR) y Aprendizaje Basado en Proyectos (ABP):

Cuadro Comparativo Aprendizaje Basado en Retos (ABR) y Aprendizaje Basado en Proyectos (ABP)		
Característica	ABR	ABP
Aprendizaje	Los estudiantes trabajan individualmente con la asesoría de formadores y tutores, para desarrollar un conocimiento y/o competencia en los temas de estudio. El reto es el detonador para la obtención de la competencia y/o conocimiento, mediante el uso de las tecnologías seleccionadas	Los estudiantes construyen su conocimiento y desarrollan competencias a través de la solución de una problemática o situación específica. En el caso del programa de formación de Campus, se refiere a la solución de una situación o problemática particular, que implica el planteamiento y desarrollo de un proyecto, en el cual se aplican los conocimientos adquiridos, en técnicas, metodologías y herramientas en el aspecto específico del desarrollo de software.
Enfoque	Enfrenta a los estudiantes a situaciones particulares pequeñas de manejo de información, que demanda una solución mediante la aplicación de técnicas, métodos y tecnologías seleccionadas.	Enfrenta a los estudiantes a situaciones complejas de manejo de información, que demanda una solución mediante la aplicación de técnicas, métodos, metodologías y herramientas para el desarrollo de software en ambiente WEB, es decir, la construcción de aplicaciones o sistemas de información.
Producto	Los estudiantes generan una solución que resulta en una acción concreta. En el programa de formación de Campus, normalmente son programas, diseños, esquema, modelos, etc.	Los estudiantes generan un producto, que soluciona una necesidad de manejo de información que beneficia a un sector de la población. En el caso del programa de formación de Campus, se produce una aplicación en ambiente WEB.
Proceso	Los tripulantes analizan, desarrollan y ejecutan la	Los estudiantes trabajan con un proyecto asignado, que



	<p>mejor solución para abordar el reto propuesto, que sea visible y medible. En el programa de formación En el programa de formación de Campus, normalmente son programas, diseños, esquema, modelos, etc..</p>	<p>genere productos para verificar su aprendizaje y desarrollo de competencias. En el programa de formación de Campus, el producto es una aplicación que beneficia a un sector de la población y podría llegar a ser el inicio de un emprendimiento.</p>
<p>Rol del profesor</p>	<p>Coach, instructor y orientador en métodos, técnicas y herramientas. El formador orienta las temáticas del ciclo, a través del aprender haciendo, apoya el conocimiento y dominio En el programa de formación de Campus..</p>	<p>Facilitador y administrador de proyectos. En el programa de formación de Campus, se convierte en un gestor de proyectos, desde el punto del cliente, para verificar el cumplimiento de sus objetivos.</p>



CICLOS DEL PROGRAMA DESARROLLO FULLSTACK

CICLO 1: FUNDAMENTOS DE PROGRAMACIÓN

En este módulo se establecen los principios fundamentales de la programación, así como los elementos y conceptos, de los programas computacionales, los cuales son de gran importancia para desarrollar habilidades de pensamiento abstracto necesario para la solución de problemas cotidianos haciendo uso de algoritmos.

Aspecto	Horas
Competencias digitales en desarrollo de software y programación	60
Acompañamiento con tutores (Review)	30
Competencias en una segunda lengua (inglés)	15
Desarrollo de habilidades blandas	15
TOTAL	120

Competencia

Aplica el pensamiento estructurado y lógico, dentro del marco del pensamiento computacional, para la solución de problemáticas o situaciones de manejo de información, utilizando el paradigma de la programación estructurada con eficiencia y responsabilidad.

Resultados de aprendizaje

- Desarrolla el pensamiento computacional, lógico y estructurado a través de la elaboración de algoritmos y diagramas de flujo, que solucionen problemas de manejo de información de manera eficiente.
- Interpreta los conceptos de programación de computadores en lo concerniente con la identificación de variables, manejo de los tipos de datos y operadores aritméticos, relacionales y lógicos de forma autocrítica.
- Reconoce la metodología para la solución de problemas de manejo de información mediante las etapas el análisis, diseño y construcción.



- Construye códigos en lenguaje Python, utilizando estructuras de control condicional de manera eficiente.
- Construye códigos en lenguaje Python, utilizando estructuras de control iterativas de manera eficiente.
- Identifica funciones en una situación de manejo de información y construye su código en lenguaje Python de manera eficiente.
- Construye códigos en lenguaje Python, utilizando estructuras de datos, tales como, arreglos y colecciones de manera eficiente.
- Maneja archivos de texto y JSON a través de los comandos e instrucción del lenguaje Python con responsabilidad.

Ejes Temáticos

Introducción a la programación.

- Pensamiento Computacional: lógico estructurado.
- Variables y constantes
- Tipos de Datos
- Operadores: Aritméticos, relacionales y lógicos
- Metodología solución de problemas de manejo de información.

Estructuras de Control

- Condicional: Simple y Anidado (IF)
- Iterativas o ciclos:
 - Controladas por cantidad (FOR)
 - Controladas por condición (WHILE)

Modularidad: Funciones

- Conceptualización
- Estructura
- Aplicaciones

Estructura de Datos

- Conceptualización
- Arreglos
- Colecciones

Persistencia de Datos

- Conceptualización
- Archivos de texto
- Archivos JSON

Plan de Aula

Semana	Unidad de Aprendizaje	Contenido
1	Introducción a la Programación	<p>Introducción a la programación.</p> <p>Pensamiento Computacional: lógico estructurado.</p> <p>Variables y constantes</p> <p>Tipos de Datos</p> <p>Operadores: Aritméticos, relacionales y lógicos</p> <p>Metodología solución de problemas de manejo de información.</p>
	Estructuras de Control	<p>Condicional: Simple y Anidado (IF)</p> <p>Formateo de las salidas</p>
2	Estructuras de Control	<p>Iterativas o ciclos:</p> <ul style="list-style-type: none"> Controladas por cantidad (FOR) Controladas por condición (WHILE) <p>Excepciones y validaciones</p>
	Modularidad: Funciones	<p>Marco conceptual de las funciones</p> <p>Estructura de una función</p> <p>Aplicaciones con funciones</p>
3	Estructura de Datos	<p>Marco conceptual de las estructuras de datos</p> <p>Arreglos</p> <p>Colecciones</p>

	Persistencia de Datos	Marco conceptual Persistencia Archivos de texto Archivos JSON Reto único
--	-----------------------	--

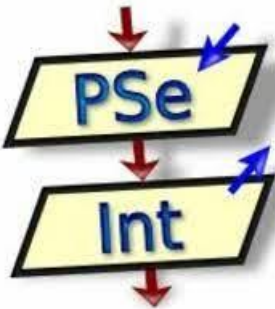
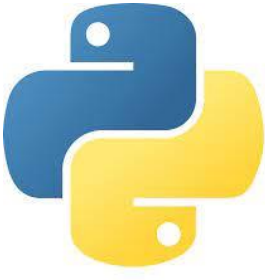
Evaluación y calificación

La evaluación del aprendizaje, integra la valoración de los retos desarrollados durante el curso. La calificación se realiza en una escala de CERO (0,0) a CINCO (5,0), para aprobar satisfactoriamente el ciclo el beneficiario debe obtener una nota igual o superior a TRES (3,5). Además, se puede manejar una nota de seguimiento al proceso del ciclo como talleres, quices, aportes en clase, etc.

$$\text{Nota Definitiva} = \text{Nota Seguimiento} * 0.1 + \text{Nota Reto único} * 0.9$$

Tecnologías

Para el desarrollo del ciclo de Fundamentos de Programación en la ruta de aprendizaje definida para esta iteración se definen las siguientes tecnologías:

Pensamiento algorítmico	Lenguaje de Programación
	



PSeINT	Python
--------	--------

Recursos – Material de apoyo

Cada estudiante, campers tendrá acceso a la plataforma donde tendrá a su disposición material de apoyo durante cada módulo.

CICLO 2: PROGRAMACIÓN BÁSICA

En el campo de acciones de la industria, es común encontrar problemas que se pueden resolver mediante el uso de programas de computadoras, estas soluciones requieren de conceptos más complejos que los estudiados en el ciclo de fundamentos de programación y obedecen a la integración del paradigma orientado a objetos y manejo de bases de datos como estrategia de resolución de problemas de manejo de información, junto con la implementación de un repositorio de código y control de versiones. Este ciclo está diseñado para madurar los conceptos de analítica, lógica, determinación de pasos de un algoritmo, planteados en la asignatura anterior, consolidando buenas prácticas de programación y diseño de soluciones sobre procesos más complejos.

Aspecto	Horas
Competencias digitales en desarrollo de software y programación	240
Acompañamiento con tutores (Review)	120
Competencias en una segunda lengua (inglés)	60
Desarrollo de habilidades blandas	60
TOTAL	480



Competencia

Aplica los principios, técnicas, metodologías y herramientas del paradigma del diseño y programación orientada a objetos para la solución de problemáticas de manejo de información utilizando las estructuras de datos, base de datos, desarrollo Web y control de versiones con responsabilidad y eficiencia.

Resultados de aprendizaje

- Interpreta la concepción del diseño WEB, a través de la aplicación de método y tecnologías con autocrítica de manera eficiente
- Construye páginas web, mediante la utilización de la tecnología HTML con responsabilidad.
- Mejora el formato de una página web, mediante la utilización de la tecnología CSS con eficiencia.
- Aplica Framework Bootstrap para mejorar la presentación y formato de una página WEB con eficiencia.
- Interpreta la concepción de los repositorios de código y control de versiones para el manejo colaborativo en el desarrollo de aplicaciones de software de manera responsable
- Maneja las aplicaciones a través de un repositorio de código y control de versiones, con autocritica de manera eficiente.
- Configura el manejo del control de versiones en una IDE como Visual Studio Code utilizando la extensión GitGraph.
- Construye códigos en lenguaje JavaScript, utilizando ambiente WEB de manera eficiente.
- Construye códigos en lenguaje TypeScript, utilizando ambiente WEB de manera eficiente.
- Construye códigos en lenguaje PHP, utilizando la programación estructurada de manera eficiente
- Interpreta los conceptos básicos de la programación orientada a objetos en lo referente a clase, objeto, constructores, encapsulamiento, visibilidad y modificadores de clase a partir de la abstracción de un contexto de forma eficiente.
- Interpreta los conceptos avanzados de la programación orientada a objetos en lo referente a herencia, polimorfismo, relaciones entre clases, colecciones de manera eficiente.
- Construye códigos utilizando la programación orientada a objetos en lenguaje TypeScript.
- Construye códigos utilizando la programación orientada a objetos en lenguaje PHP.



- Interpreta los conceptos básicos del diseño orientado a objetos en lo referente con diagramas de clases manejando relaciones de agregación y composición de manera eficiente.
- Interpreta la concepción de las bases de datos relacionales, como elemento de almacenamiento de una aplicación de software, en sus aspectos de modelamiento para la generación de modelos entidad- relación de manera responsable.
- Construye esquemas de base de datos, utilizando el SQL en su aspecto del Lenguaje de Definición de Datos (DDL), utilizando sistemas administradores de bases de datos como MySQL.
- Manipula la información de la base de datos, utilizando el SQL en su aspecto del Lenguaje de Manipulación de Datos (DML), utilizando sistemas administradores de bases de datos como MySQL.
- Construye códigos SQL a nivel de DDL y DML utilizando el sistema administrador de bases de datos MySQL.
- Construye códigos SQL a nivel de DDL y DML utilizando el sistema administrador de bases de datos PostgreSQL
- Interpreta la concepción de las bases de datos no relacionales NoSQL, como elemento de almacenamiento de masivo de información, en sus aspectos de modelamiento y manipulación de manera responsable.
- Construye y manipula base de datos no relaciones, a través de tecnologías como MongoDB, con autocritica de manera eficiente

Ejes Temáticos

Desarrollo WEB.

- Conceptualización sobre Diseño Web
- HTML.
 - .
 - Estructura básica de un documento HTML.
 - Etiquetas de texto Etiquetas de párrafo
 - Etiquetas HTML5 (footer, header, section, nav)
 - Listas
 - Contenedores (Div, Span, IFrame) Tablas.
 - Multimedia (imágenes, Audio y Video)
- CSS
 - Introducción Metodología BEM
 - Unidades de medida (Absolutas y Relativo)
 - Márgenes Padding
 - Definición de reglas de estilo – Maquetación básica
- Bootstrap
 - Conceptualización

- Elementos

Control de Versiones

- Conceptualización
- Configuración básica.
- Inicialización de repositorios
- Modificadores aplicados al commit Trazabilidad en commit
- Gestión y administración de ramas en GitHub
- Commit Documentation
- Despliegue de versiones
- Conceptos básicos de GitGraph
- Instalación y configuración en visual studio code.
- Documentación GitGraph

JavaScript, TypeScript, PHP

- Conceptualización JavaScript
- Tipos de Datos y operadores en JavaScript
- (Ternario) Template String en JavaScript
- Estructuras condicionales y Estructuras iterativas(Ciclos) en JavaScript
- Procedimientos Funciones Función flecha JavaScript
- Lambda function JavaScript
- Arrays Object Array en JavaScript
- Funciones array Desestructuración en JavaScript
- Map y Set en JavaScript
- D.O.M en JavaScript
- LocalStorage
- Módulos
- Objetos literales
- Desestructuración
- Promesas, Fetch Api, Ternarios, Async, Await
- TypeScript
- PHP

Programación Orientada a Objetos

- Clases, Objetos, Constructores, Visibilidad, modificadores de Clase.
- Encapsulamiento, Getter and setter
- Polimorfismo Herencia, Interfaces y Clases abstractas.
- Relaciones entre clases. Inducción al diseño orientado a objetos

Base de Datos Relacionales

- Teoría de conjuntos

- Marco conceptual.
- Diseño base de datos (Normalización)
- SQL: Conceptualización.
- SQL – Lenguaje de Definición de Datos (DDL).(create database, create table, alter table, drop table, describe)
- SQL – Lenguaje de Manipulación de Datos (DML) (insert, update, delete y select)

Base de Datos No relacionales

- Conceptualización.
- Manejo de bases de datos No relacionales. (Creación y Operaciones)

Plan de Aula

Semana	Unidad de Aprendizaje	Contenido
1-2	HTML	<p>Conceptualización sobre Diseño Web.</p> <p>Estructura básica de un documento HTML.</p> <p>Etiquetas de texto Etiquetas de párrafo</p> <p>Etiquetas HTML5 (footer, header, section, nav)</p> <p>Listas</p> <p>Contenedores (Div, Span, IFrame) Tablas.</p> <p>Multimedia (imágenes, Audio y Video)</p>
	CSS	<p>Introducción Metodología BEM</p> <p>Unidades de medida (Absolutas y Relativo)</p> <p>Márgenes Padding</p> <p>Definición de reglas de estilo – Maquetación básica</p>

	Bootstrap	<p>Conceptualización Framework Bootstrap</p> <p>Elementos</p>
3	Control de Versiones	<p>Conceptualización</p> <p>Repositorio de código y control de versiones</p> <p>Configuración básica.</p> <p>Inicialización de repositorios</p> <p>Modificadores aplicados al commit Trazabilidad en commit</p> <p>Gestión y administración de ramas en GitHub</p> <p>Commit Documentation</p> <p>Despliegue de versiones</p> <p>Conceptos básicos de GitGraph</p> <p>Instalación y configuración en visual studio code.</p> <p>Documentación</p> <p>Reto 1</p>
4-5-6-7	JavaScript, TypeScript, PHP	<p>Conceptualización JavaScript</p> <p>Tipos de Datos y operadores en JavaScript</p> <p>(Ternario) Template String en JavaScript</p>

		<p>Estructuras condicionales y Estructuras iterativas(Ciclos) en JavaScript</p> <p>Procedimientos Funciones Función flecha en JavaScript</p> <p>Lambda function en JavaScript</p> <p>Arrays Object Array en JavaScript</p> <p>Funciones array Desestructuración en JavaScript</p> <p>Map y Set en JavaScript</p> <p>D.O.M en JavaScript</p> <p>LocalStorage</p> <p>Módulos</p> <p>Objetos literales</p> <p>Desestructuración</p> <p>Promesas, Fetch Api, Ternarios, Async, Await</p> <p>TypeScript</p> <p>PHP</p>
--	--	---

8-9	Programación Orientada a Objetos	<p>Clases, Objetos, Constructores, Visibilidad, modificadores de Clase.</p> <p>Encapsulamiento, Getter and setter</p> <p>Polimorfismo Herencia, Interfaces y Clases abstractas.</p> <p>Relaciones entre clases (Inducción al diseño orientado a objetos)</p> <p>Reto 2</p>
10-11	Base de Datos	<p>Teoría de conjuntos</p> <p>Marco conceptual sobre Base de Datos</p> <p>Diseño base de datos (Normalización)</p> <p>SQL: Conceptualización.</p> <p>SQL – Lenguaje de Definición de Datos (DDL).(create database, créate table, alter table, drop table, describe)</p> <p>SQL – Lenguaje de Manipulación de Datos (DML) (insert, update, delete y select)</p>
12	Base de Datos No relacionales	<p>Conceptualización Base de datos no relacionales</p> <p>Manejo de bases de datos No relacionales. (Creación y Operaciones) con MongoDB</p> <p>Reto 3</p>

Evaluación y calificación

La evaluación del aprendizaje, integra la valoración de los retos desarrollados durante el curso y el avance en el proyecto planteado en el programa. La calificación se realiza en una escala de CERO (0,0) a CINCO (5,0), para aprobar satisfactoriamente el ciclo el beneficiario debe obtener una nota igual o superior a



TRES (3,5). Además, se puede manejar una nota de seguimiento al proceso del ciclo como talleres, quices, aportes en clase, etc.

$$\text{Nota Definitiva} = \text{Nota Seguimiento} \times 0.1 + \text{Promedio Nota Retos} \times 0.9$$

Tecnologías

Para el desarrollo del ciclo de Programación Básica en la ruta de aprendizaje definida para esta iteración se definen las siguientes tecnologías:

Desarrollo WEB		
HTML 	CSS 	
Control de versiones		



Programación Orientada a objetos



Base de Datos



Recursos – Material de apoyo

Cada estudiante, campers tendrá acceso a la plataforma donde tendrá a su disposición material de apoyo durante cada módulo.



CICLO 3: DESARROLLO FULLSTACK.

En este módulo, se estudian los principios del diseño y construcción aplicaciones de software a través de las tecnologías Backend y Frontend, que permitirán el desarrollo de aplicaciones exitosas y de alta aceptación por parte de los usuarios.

Aspecto	Horas
Competencias digitales en desarrollo de software y programación	340
Acompañamiento con tutores (Review)	170
Competencias en una segunda lengua (inglés)	85
Desarrollo de habilidades blandas	85
TOTAL	680

Competencia

Aplica los principios, técnicas, metodologías y herramientas de la ingeniería del software para el desarrollo de aplicaciones WEB, utilizando repositorios de código y diseño Backend - Frontend con autocrítica, responsabilidad y eficiencia.

Resultados de aprendizaje

.



- Identifica el modelo de ciclo de vida para el desarrollo de software, roles en el desarrollo de software y requerimiento funcional y no funcional de manera responsable.
- Interpreta la concepción del Backend - Frontend dentro del marco de la arquitectura de microservicios, para el diseño de aplicaciones WEB, con eficiencia y autocrítica.
- Construye aplicaciones Backend mediante la utilización responsable y eficiente del Framework Node JS.
- Construye aplicaciones Backend mediante la utilización responsable y eficiente del Framework Laravel.
- Construye aplicaciones Frontend mediante la utilización responsable y eficiente del Framework React JS
- Construye aplicaciones Frontend mediante la utilización responsable y eficiente del Framework VUI

Ejes Temáticos

Marco conceptual básico para el desarrollo de software.

- Ciclo de vida del software.
- Roles en desarrollo de software.
- Definición de Requerimiento funcional y no funcional.
- Presentación del proyecto de software a desarrollar

Desarrollo Backend.

- Marco Conceptual Backend
- Arquitectura de microservicios
- Estructura de funcionamiento del Backend - Frontend
- Frameworks Node JS.
 - Configuración Entorno de desarrollo (Node JS, NPM, CLI Framework de desarrollo)
 - Generación de proyecto Backend usando Node JS.
 - Estructura básica del Backend.
 - Controladores, Repositorio de servicios, Rutas, Verbos HTTP, Conexión a BD y Ejecución del servidor.
 - Consumo de servicios REST usando Postman o Insomnia.
- .Frameworks Laravel.
 - Configuración inicial (rutas, vistas, controladores, sistema de autenticación, Git, GitHub)



- Diseño (integrando assets, sistema de plantillas Blade, estructura de plantilla e integración al proyecto, definiendo menú de navegación, plantilla vista registro)
- Modelo de datos (migraciones y modelos)
- Entidades del proyecto y datos (entidad-producto, ORM Eloquent, Seeders, inyección de datos, Tinker).
- Gestión de datos (Realizar CRUD, paginación, formularios de registro, edición y eliminación, validación de datos).
- Roles y permisos
- Imágenes y gestión de archivos.
- Características finales.

Desarrollo Frontend

- Framework REACT JS.
 - Conceptos básicos y Arquitectura SPA
 - SPA vs MPA
 - Configuración SPA usando Frameworks de Desarrollo React JS.
 - Estructura de proyectos SPA
 - Componentes en SPA (Estructura, programación, definición de variables, objetos)
 - Integración de Componentes y vista (HTML)
 - Configuración de dependencias FrontEnd
 - Consumo de servicios desde el FrontEnd
- Frameworks VUI.
 - Introducción al VUI.
 - Vista (Templates. Atributos, expresiones, directivas, manejo de datos, validaciones, scopes entrega de datos a componentes),
 - Comportamientos (Métodos, herencia, Mixins, eventos, formularios, validaciones, persistencia)
 - Animaciones (estilos, clases CSS, directivas, transiciones, enviar datos AXIOS,

Plan de Aula

Semana	Unidad de Aprendizaje	Contenido
1	Marco conceptual básico para el desarrollo de software.	Ciclo de vida del software. Roles en desarrollo de software. Definición de Requerimiento funcional y no funcional.

		Presentación del proyecto de software a desarrollar
	Backend	<p>Marco Conceptual Backend</p> <p>Arquitectura de microservicios</p> <p>Estructura de funcionamiento del Backend - Frontend</p>
2-8	Desarrollo Backend. – Frameworks Node JS	<p>Configuración Entorno de desarrollo (Node JS, NPM, CLI Framework de desarrollo)</p> <p>Generación de proyecto Backend usando Node JS.</p> <p>Estructura básica del Backend.</p> <p>Controladores, Repositorio de servicios, Rutas, Verbos HTTP, Conexión a BD y Ejecución del servidor.</p> <p>Consumo de servicios REST usando Postman o Insomnia</p>
9-13	Desarrollo Backend.- Frameworks Laravel	<p>Configuración inicial (rutas, vistas, controladores, sistema de autenticación, Git, GitHub)</p> <p>Diseño (integrando assets, sistema de plantillas Blade, estructura de plantilla e integración al proyecto, definiendo menú de navegación, plantilla vista registro)</p> <p>Modelo de datos (migraciones y modelos)</p> <p>Entidades del proyecto y datos (entidad-producto, ORM Eloquent, Seeders, inyección de datos, Tinker).</p> <p>Gestión de datos (Realizar CRUD, paginación, formularios de registro, edición y eliminación, validación de datos).</p>

		<p>Roles y permisos</p> <p>Imágenes y gestión de archivos.</p> <p>Características finales</p>
14-15	Desarrollo Frontend – Frameworks React JS	<p>Conceptos básicos y Arquitectura SPA</p> <p>SPA vs MPA</p> <p>Configuración SPA usando Frameworks de Desarrollo React JS.</p> <p>Estructura de proyectos SPA</p> <p>Componentes en SPA (Estructura, programación, definición de variables, objetos)</p> <p>Integración de Componentes y vista (HTML)</p> <p>Configuración de dependencias FrontEnd</p> <p>Consumo de servicios desde el FrontEnd</p>
16-17	Desarrollo Frontend – Frameworks VUI	<p>Introducción al VUI.</p> <p>Vista (Templates. Atributos, expresiones, directivas, manejo de datos, validaciones, scopes entrega de datos a componentes),</p> <p>Comportamientos (Métodos, herencia, Mixins, eventos, formularios, validaciones, persistencia)</p> <p>Animaciones (estilos, clases CSS, directivas, transiciones, enviar datos AXIOS)</p>

Evaluación y calificación





La evaluación del aprendizaje se realizará a través de los resultados del desarrollo del proyecto, que corresponde a una aplicación de software. La calificación se realiza en una escala de CERO (0,0) a CINCO (5,0), para aprobar satisfactoriamente el ciclo el beneficiario debe obtener una nota igual o superior a TRES (3,5). Además, se puede manejar una nota de seguimiento al proceso del ciclo como talleres, quices, aportes en clase, etc.

$$\text{Nota Definitiva} = \text{Nota Seguimiento} * 0.1 + \text{Promedio Nota Proyecto} * 0.9$$

Tecnologías

Para el desarrollo del ciclo de Desarrollo FullStack en la ruta de aprendizaje definida para esta iteración se definen las siguientes tecnologías:

Backend	
	
Frontend	



Vue.js



React JS

Recursos – Material de apoyo

Cada estudiante, campers tendrá acceso a la plataforma donde tendrá a su disposición material de apoyo durante cada módulo.