**DESARROLLO FULL STACK Y APLICACIONES HÍBRIDAS**

**Objetivo**

Capacitar profesionales, técnicos y tecnólogos en competencias para proponer soluciones y escenarios tecnológicos por medio de prototipos, procesamiento de datos y la construcción de aplicaciones para plataformas móviles desde componentes Web, aprovechando las herramientas que nos ofrece los actuales frameworks del mercado.



**Metodología**

Usando el **Aprendizaje Basado en Proyectos** (ABP o PBL, Project-based learning) nos permite aproximarnos a casos reales, basándonos en la necesidad de cambiar el paradigma de los procesos formativos, teniendo claro el porqué, para qué y las necesidades en la vida de cada una de las competencias adquiridas, consolidando un aprendizaje con sentido.

El método consiste en la realización de un proyecto, normalmente de cierta envergadura en un grupo diversificado e integral. El proyecto será analizado previamente por el profesor para asegurarse que el alumno tiene lo necesario para resolverlo, y que en su resolución desarrollara todas las destrezas que se desean. El desarrollo del proyecto iniciará con una problemática básica a resolver. Esta debe basar su solución en el análisis de la problemática, la investigación y el pensamiento crítico, además de la implementación de la tecnologías y conocimientos adquiridos para generar productos eficientes como resultado del proceso.



**Etapas de formación:**

· **Etapa Lectiva**: Periodo de tiempo en el cual el alumno recibe la formación integral en las aulas que facilitan el logro de resultados de aprendizaje.

· **Etapa Productiva**: periodo de tiempo en el cual el alumno aplica los conocimientos, habilidades y destrezas desarrolladas en la etapa lectiva.

· **Evaluación de aprendizaje**: Conjunto de juicios emitidos con base en los resultados de aprendizaje y criterios de valoración sobre los logros del alumno, sus conocimientos y habilidades.

· **Evidencia de aprendizajes**: Pruebas manifestadas de aprendizajes recogidas en el proceso de formación; son recolectadas por el instructor utilizando métodos o técnicas de evaluación seleccionadas según sean sus evidencias de conocimiento para que permita un posterior desempeño laboral y social.

· **Evidencias de conocimiento**: manifestación ciertas y claras sobre el aprendizaje del alumno relacionadas con el saber, el comprender, el hacer, esto incluye el conocimiento de hechos y de los principios y la manera de aplicar los conocimientos en situaciones nuevas.

· **Evidencias de producto**: resultado que se tiene en el desarrollo durante el curso **(prototipo)**

**Prerrequisitos:**

Se requiere del estudiante conocimientos básicos de los lenguajes base para el desarrollo de tecnologías web.

**Evaluación de conocimientos**

El contenido de la evaluación serán compuestos por los siguientes ítems:

* Tecnologías web básicas
* Lenguajes de programación
* Funcionamiento de plataformas web

**Contenido del programa**

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulo** | **Contenido** | **Herramientas** | **Horas** |
| **Nivelación (Opcional)** | Desarrollo Web Básico  Fundamentos de programación web  HTML, CSS | Sublime Text 3 | **4 h** |
| **Prototipo inicial** | Planteamiento de objetivos de un producto  Planear prototipos  Diseño de prototipos en papel  Diseño de prototipo digital  Evaluación de eficiencia en prototipo  Patrones animados de diseño de apps | AdobeXD  WebFlow | **20h** |
| **Programación Web Módulo I** | JavaScript  Crear Funciones, evento y Objetos  Diseño de algoritmos  Git y Github  Base de datos SQL (Postgres) | Nodejs  Git - Github  PostgresQL | **15h** |
| **Programación Web Módulo II** | Comprender las partes principales de RUBY  Manejar estructura de datos  Flujo y control de un programa  Ruby on Rails  Arquitectura MVC  Conexión a Base de Datos  API REST | Ruby on Rails  Postman  Nginx | **20h** |
| **Prototipo Avanzado** | Conexión Backend - FrontEnd  Prototipo Funcional  Desarrollo Beta  Despliegue de datos Google Cloud | Google Cloud  Google Console | **20hrs** |
| **Desarrollo de Aplicaciones híbridas** | Typescript  Ionic 3 – Angular 5  Arquitectura de una Apps  Emuladores  Uso de plugins Nativos (Camara - Agenda)  Notificaciones Push  Compilación ANDROID, IOS  Publicación en tiendas | Angular 5  Ionic 3  Android SDK  Genymotion | **28hrs** |

**Diseño y Revisión del programa:**

**Jhon Cortés**: Ingeniero de CTO en HomeSquad con más de 8 años de experiencia en tecnologías Webs y móviles. Durante su experiencia como formador diseñó y dirigió el programa de capacitaciones en tecnología móviles para:

* Agencia Alemana para la Cooperación Internacional (Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH) en El Salvador.
* Programa de desarrollo móvil, Manizales Vive Digital.
* Capacitación en Nuevas tecnologías para comunidades vulnerables en el estado de Chiapas - México.

**Javier Jaimes:** Ingeniero de Software Full Stack en ZagaLabs, con 10 años de experiencia en desarrollo grandes compañías como VoiceBunny del grupo Torrenegra Labs, Globant, PSL, entre otras. Durante su trayectoria como formador ha representado a Colombia en el Startup Young de Malasia organizado por el Gobierno Estadounidense, y a diseñó y dirigió el programa Co-crea Colombia para el Banco Interamericano de Desarrollo (BID).

**Juan David Zapata**: Co-founder WeHostels (Adquirida por Student Universe). Full Stack developer en indeed.com compañia estadounidense disponible en más de 50 países y en más de 28 idiomas.

**Carlos Rojas:** Google Product Strategy Expert, desarrollador Ionic para Globant una de las compañías de software más grandes a nivel mundial.

**Material**

* Repositorio con los ejercicios de cada clase
* Material Guia de clases
* Ejercicios de cada módulo

**Docentes**

**Perfil Docente Front End**

Desarrollador Senior con más de 4 años de experiencia con conocimientos en:

* Tecnologías javaScript, Angularjs y Ionic.
* Conocimientos en consumo de datos mediante API’s
* Análisis y Diseño de producto

**Perfil Docente Back End**

Desarrollador senior con más de 4 años de experiencia con conocimientos en:

* Tecnologías Ruby, Ruby on Rails
* API REST
* Bases de datos relacionales y diseño de esquemas
* Análisis y Diseño de producto

|  |  |  |
| --- | --- | --- |
| **Prototipado inicial (20 horas)** | | |
| **Clase 1** | Temario | Introducción al Programa de formación.   * Metodología * Socialización temario * Objetivos del programa * Prototipado básico   + Prototipos (Productos Digitales).   + Niveles de fidelidad de un prototipo.   + Wireframes en Papel.   + Wireframes en Digital. |
|
| Herramientas | Gliffy(gliffy.com) |
| Actividad | Crear Sketch y wireframe de la web Woven azine.com) |
| **Clase 2** | Temario | 1.1 Diseño Modular  1.2 Mockups  1.3 Prototipo interactivo |
|
| Herramientas | moqups (moqups.com)  fluidui (fluidui.com) |
| Actividad | Crear un prototipo funcional Instagram |
| **Clase 3** | Temario | 2. HTML  2.1 Introducción HTML  2.2 Anatomía HTML  2.3 Estructura HTML  3. HTML + CSS  3.1 Introducción CSS  3.2 Inline Styles  3.3 selectores  3.4 .css files  3.5 Elementos anidados |
|
| Herramientas | Sublime Text |
| Actividad |  |
| **Clase 4\*** | Temario | 3.6 Box model  3.7 Colores, Tipografía  3.8 CSS Grids  3.9 Diseño adaptativo |
|
| Herramientas | Sublime Text |
| Actividad | Replicar la web de Woven (wovenmagazine.com) |
| **Clase 5** | Temario | 4. Diseño de proyecto  4.1 Presentación de Retos  4.2 Buscando Soluciones  4.3 Creatividad y teoría de pensamiento lateral  4.5 MVP |
|
| Herramientas | --- |
| Actividad | Definir proyecto  Sketch del Proyecto |
| **Programación Web (16 horas)** | | |
| **Clase 1** | Temario | 1. Conceptos básicos    1. Terminal o CLI    2. Servidores    3. Dominios    4. Protocolo http y https 2. Entorno de desarrollo    1. Instalar nodeJs 3. Git & Github    1. Crear cuenta Github    2. init, create, clone, commit, pull, push |
|
| Herramientas | Github, CLI |
| Actividad | Crear cuenta en Github y su primer repositorio |
| **Clase 2** | Temario | 4. JavaScript  4.1 Tipo de datos  4.2 Variables, Constantes  4.3 Operaciones matemáticas  4.4 Estructuras de Control  4.5 if/else  4.6 Operadores lógicos  4.7 Switch  4.8 Funciones |
|
| Herramientas | Consola, Sublime Text |
| Actividad | Subir ejercicios de la clase a Github |
| **Clase 3** | Temario | 4.9 Scope  4.10 Arrays  4.11 Ciclos e Iteradores  4.12 Objetos y Clases  4.13 Módulos  4.14 Procesos Asíncronos y Síncronos  4.14 Peticiones HTTP |
|
| Herramientas | Github, JS |
| Actividad | Conectar con jsonplaceholder.typicode.com y imprimir el resultado.  Subir ejercicios de la clase a Github. |
| **Clase 4** | Temario | 5 Bases de Datos  5.1 Instalar Postgres  5.2 Propósito general de BD  5.3 Modelo Entidad Relación  5.4 SQL  5.5 Estructura de un Query  5.6 Insertar Registros  5.5 Joins  5.6 Usuarios  5.7 Roles |
|
| Herramientas | Postgres |
| Actividad | Crear modelo entidad relación de Proyecto |
| **Programación Web (20 horas)** | | |
| **Clase 1** | Temario | 1. Ruby   1.2 Instalar Ruby  1.3 Tipos de datos  1.4 Variables  1.5 Operaciones Matemáticas  1.6 Estructuras de control  1.7 Todos son Objetos  1.8 Estructuras de control  1.9 array & Hashes  1.10 Ciclo y iteradores  1.11 Blocks, Procs y Lambdas  2. Programación Orientada a Objetos |
|
| Herramientas | Ruby - CLI - Editor de Texto |
| Actividad | - |
| **Clase 2** | Temario | 2 Ruby on Rails  2.1 Instalar Rails  2.2 bundle & gem  2.3 Estructura de Archivos  2.4 Entornos de Desarrollo  2.5 MVC  2.6 Controladores, rutas y vistas  2.7 Scaffold  2.8 Métodos de petición HTTP |
|
| Herramientas | Ruby - CLI - Editor de Texto - Rails |
| Actividad | Generar proyecto con landing page  Actualizar en Github |
| **Clase 3** | Temario | 3 Conexión DB  3.1 Comando Rake  3.2 Configurar conexión  3.3 Modelos  3.4 Migraciones  3.5 Asociaciones |
|
| Herramientas | Ruby - CLI - Editor de Texto - Rails |
| Actividad | Agregar Migraciones necesarias al proyecto  Actualizar Github |
| **Clase 4** | Temario | 4. Autenticación y Autorización  4.1 Modelo de Usuarios  4.2 Registro de usuarios  4.3 Ingreso de usuarios  4.4 Salida  4.5 current\_user (Sessions - Cookies)  4.6 Filtros |
|
| Herramientas | Ruby - CLI - Editor de Texto - Rails |
| Actividad | Agregar Autenticación al proyecto  Actualizar Github |
| **Clase 5** | Temario | 5. API  5.1 Json  5.2 Autenticación vía Token |
|
| Herramientas | Ruby - CLI - Editor de Texto - Rails |
| Actividad | Agregar Endpoints al proyecto  Actualizar Github |
| **Prototipo avanzados (16 horas)** | | |
| **Clase 1** | Temario | 1 Autenticación y autorización API  1.2 refrescar Token  1.3 Guardar Session |
|
| Herramientas | Ruby - CLI - Editor de Texto - Rails |
| Actividad | Agregar Endpoints al proyecto  Actualizar Github |
| **Clase 2** | Temario | 2. Notificaciones  2.1 Conexión a servidor de Emails  2.2 Plantilla de emails |
|
| Herramientas | Ruby - CLI - Editor de Texto - Rails |
| Actividad | Agregar Notificaciones al proyecto  Actualizar Github |
| **Clase 3** | Temario | 3. Conexiones SSH  3.1 Administrar llaves públicas  3.2 Comandos basicos Linux  3.3 Instalar Nginx |
|
| Herramientas | SSH - CLI |
| Actividad | Prepara servidor |
| **Clase 4** | Temario | 4. Capistrano  4.1 Configuración  4.3 Deploy |
|
| Herramientas | SSH - IDE - GitHub |
| Actividad | Publicar API |
| **Aplicaciones híbridas (32 horas)** | | |
| **Clase 1** | Temario | 1. TypeScript  1.1 Sintaxis  1.2 Angular 5  1.3 Ionic 3  1.4 Aplicaciones Híbridas  1.5 Diferencia Nativo/Híbrido  1.6 Crear Proyecto Ionic  1.7 Estructura de carpetas |
|
| Herramientas | Ionic - angular - Github |
| Actividad | Iniciar mi primera APP  Crear nuevo repositorio Github |
| **Clase 2** | Temario | 2. Componentes  2.1 Botones  2.2 Listas  2.3 Alertas  2.4 Tabs  2.5 Slides  2.6 Navegación  2.7 Listas  2.8 Menús laterales |
|
| Herramientas | Ionic - angular - Github |
| Actividad | Crear formularios y vista principal  Actualizar Repositorio Github |
| **Clase 3** | Temario | 3. Servicios y Providers |
|
| Herramientas | Github - Ionic 3 |
| Actividad | Actualizar Github |
| **Clase 4** | Temario | 4. Autenticación  4.1 Conexión a Servidor  4.2 LocalStorage  4.3 Sessions |
|
| Herramientas | Ionic - angular - Github |
| Actividad | Agregar autenticación a nuestra aplicación  Actualizar Repositorio Github |
| **Clase 5** | Temario | 5. Notificaciones  5.1 Notificaciones push  5.2 notificaciones toast |
|
| Herramientas | Ionic - angular - Github |
| Actividad | Agregar notificaciones a nuestra aplicación  Actualizar Repositorio Github |
| **Clase 6** | Temario | 6. Recursos del teléfono  6.1 Cámara  6.2 GPS  6.3 Agenda |
|
| Herramientas |  |
| Actividad |  |
| **Clase 7** | Temario | Desarrollo de Proyecto  Correr aplicaciones en android  Correr aplicaciones en un emulador android  Correr aplicacion en un dispositivo IOS  Correr aplicaciones en un emulador iOS |
|
| Herramientas | Ionic - angular - Github |
| Actividad | Actualizar Repositorio Github |
| **Clase 8** | Temario | Registro en Tiendas  Publicación APP |
|
| Herramientas | Ionic - angular - Github |
| Actividad | Prototipo Final |