

PLAN DE ESTUDIO

JUNIORITY

PREUNIVERSITARIO TECNOLÓGICO

06-01-2025

INDICE

1. Resumen Ejecutivo	6
2. Introducción	7-8
3. Contexto y Justificación del Proyecto	9
3.1. Problemas Identificados	9
3.2 Oportunidades que Aborda Juniority	10
3.3. Impacto Esperado	10
4. Objetivos del Proyecto	10
4.1. Objetivo General	10
4.2. Objetivos Específicos	11
5. Descripción General del Programa	12
5.1. Metodología de Enseñanza	12-13
5.2. Duración y Estructura Temporal	13
5.3. Enfoque Pedagógico	13-14
6. Módulos del Programa	14

INDICE

6.1. Prototipos y diseño (Figma/Miro, modelado de datos)	14
6.2. Fundamentos de SQL y bases de datos	15
6.3. Programación orientada a objetos	15
6.4. Desarrollo web con HTML, CSS y JavaScript	15
6.5. React y desarrollo frontend	16
6.6. Backend con Node.js y APIs REST	16
6.7. PostgreSQL y consultas avanzadas	16
6.8. Proyecto final	17
7. Perfil del Estudiante Egresado	17
7.1 Habilidades Técnicas	17-18
7.2 Habilidades Blandas	18
7.3 Proyección Profesional	18
8. Metodología de Evaluación	19
8.1 Evaluaciones Semanales: Enfoque Formativo	19-20

INDICE

8.2 Evaluaciones Mensuales: Enfoque Sumativo	20
8.3 Proyecto Final: Integración de Aprendizajes	20-21
9. Recursos Educativos y Herramientas	21
9.1 Plataformas Digitales	21-22
9.2 Material de Apoyo	22-23
9.3 Requisitos Técnicos	23
10. Rol del Docente	23
10.1 Funciones Principales	23-24
10.2 Interacción con los Grupos	24
10.3 Seguimiento del Progreso	24-25
11. Evaluación y Seguimiento del Proyecto	25
11.1 Indicadores Clave de Éxito (KPIs)	25
11.2 Retroalimentación de Estudiantes y Apoderados	25-26
11.3 Informes de Avance	26

INDICE

12. Plan de Comunicación	26
12.1 Comunicación con Alumnos y Apoderados	26-27
12.2 Atención de Dudas y Consultas	27
12.3 Protocolo de Comunicación	27
13. Cierre del Programa y Certificación	28
13.1 Evento de Clausura	28
13.2 Entrega de Certificados por Módulo	28-29
13.3 Certificado Final del Programa	29
13.4 Reconocimientos Especiales	29-30
14. Anexos y Contactos	30
14.1 Contacto del Equipo	30
14.2 Equipo Directivo y Coordinación	30-31
14.3 Redes Sociales Oficiales	31

1. RESUMEN EJECUTIVO

Juniority: Preuniversitario Tecnológico es un programa educativo innovador diseñado para preparar a estudiantes de enseñanza media en habilidades tecnológicas altamente demandadas en el mercado actual. Este preuniversitario tiene como objetivo principal cerrar la brecha entre la formación escolar tradicional y los requerimientos técnicos y prácticos del mundo universitario y profesional, ofreciendo un enfoque práctico, colaborativo y alineado con las tendencias tecnológicas actuales.

El programa está estructurado en **8 módulos temáticos**, que abarcan desde el diseño de prototipos y bases de datos hasta el desarrollo frontend, backend y la gestión de proyectos tecnológicos. Los estudiantes aprenderán a utilizar herramientas clave como **Figma, Miro, MySQL Workbench, NetBeans, PostgreSQL, React y Node.js**, culminando con un proyecto final integrador que refleje todas las competencias adquiridas durante el curso.

La metodología de enseñanza combina sesiones grupales donde se abordan conceptos fundamentales, con sesiones personalizadas (grupos de 3 alumnos) donde los docentes actúan como directores de proyecto, guiando y supervisando el trabajo colaborativo de los estudiantes. Este enfoque busca no solo impartir conocimientos técnicos, sino también desarrollar habilidades blandas como el trabajo en equipo, la comunicación efectiva y la resolución de problemas.

Juniority se presenta como una respuesta directa a la creciente demanda de profesionales con habilidades tecnológicas sólidas, contribuyendo a la formación de una nueva generación de talentos preparados para enfrentar los desafíos del siglo XXI.

Al finalizar el programa, los estudiantes contarán con las herramientas necesarias para desempeñarse en entornos tecnológicos, continuar su formación académica en áreas relacionadas o incluso iniciar proyectos propios.

El éxito de Juniority radica en su capacidad para inspirar, capacitar y empoderar a jóvenes talentos, facilitando su transición hacia el mundo profesional y académico con confianza y preparación.

2. INTRODUCCIÓN

En un mundo cada vez más digitalizado, la programación se ha convertido en una de las habilidades más demandadas y valiosas del mercado laboral. No solo es una herramienta clave para el desarrollo de software y aplicaciones, sino también una competencia transversal que potencia el pensamiento lógico, la resolución de problemas y la creatividad. Sin embargo, existe una brecha significativa entre los contenidos que se imparten en la educación tradicional y las habilidades prácticas requeridas para ingresar con éxito en carreras tecnológicas.

Juniority: Preuniversitario Tecnológico surge como una respuesta a esta necesidad, ofreciendo un programa educativo diseñado específicamente para estudiantes de enseñanza media con interés en adentrarse en el mundo de la programación y la tecnología. A través de un enfoque práctico y orientado a proyectos, el curso guía a los participantes desde los fundamentos del diseño de interfaces y bases de datos, hasta la implementación de aplicaciones completas mediante herramientas y lenguajes y entornos de programación modernos como **Java, JavaScript, React, y Node.js**.

Este programa no solo busca enseñar a los estudiantes a escribir código, sino también a comprender el ecosistema completo del desarrollo tecnológico, desde la planificación de una aplicación hasta su despliegue final. Además, se fomenta el desarrollo de habilidades blandas esenciales, como el trabajo en equipo, la comunicación efectiva y la capacidad de adaptarse a nuevas herramientas y entornos tecnológicos.

La estructura de **Juniority** ha sido cuidadosamente diseñada para equilibrar teoría y práctica, permitiendo que los estudiantes no solo adquieran conocimientos sólidos, sino que también los apliquen en proyectos reales que simulan desafíos del mundo profesional y universitario. Las sesiones grupales proporcionan un espacio para el aprendizaje colaborativo, mientras que las sesiones de trabajo dirigido permiten a los docentes ofrecer una guía más personalizada y orientada a los objetivos específicos de cada equipo.

En definitiva, **Juniority** no es solo un curso de programación, sino una experiencia formativa integral que prepara a los estudiantes para enfrentar con confianza y competencia su futuro académico y profesional en el apasionante mundo de la tecnología.

3. CONTEXTO Y JUSTIFICACIÓN DEL PROYECTO

La programación y el desarrollo de software se han convertido en pilares fundamentales para la innovación y el crecimiento económico, sin embargo, la mayoría de los estudiantes de enseñanza media no tienen acceso a una formación adecuada que les permita desarrollar habilidades técnicas ni explorar su potencial en áreas tecnológicas antes de ingresar a la educación superior.

Juniority: Preuniversitario Tecnológico surge como una solución innovadora para cerrar esta brecha, proporcionando una base sólida en programación, desarrollo de aplicaciones y tecnología a jóvenes con interés en carreras relacionadas con la informática y el desarrollo tecnológico. Este programa busca no solo impartir conocimientos técnicos, sino también inspirar vocaciones, reducir la incertidumbre vocacional y preparar a los estudiantes para enfrentar los desafíos académicos y profesionales que encontrarán en su futuro inmediato.

3.1. PROBLEMAS IDENTIFICADOS:

1. **Desconocimiento sobre el campo de la programación:** Muchos estudiantes no tienen claridad sobre las áreas de aplicación de la programación y sus múltiples salidas profesionales
2. **Escasa formación práctica en tecnología:** La enseñanza tradicional no suele ofrecer un enfoque práctico y orientado a proyectos en herramientas modernas de desarrollo.
3. **Falta de habilidades transversales:** Los estudiantes a menudo carecen de habilidades como el trabajo en equipo, la gestión de proyectos y la comunicación efectiva.
4. **Desafío en la transición a la educación superior:** El salto desde la enseñanza media a carreras tecnológicas puede ser abrumador para muchos jóvenes debido a la falta de conocimientos previos.

3.2 OPORTUNIDADES QUE ABORDA JUNIORITY:

- **Desarrollar habilidades técnicas desde una etapa temprana:** Proporciona a los estudiantes las bases en programación, bases de datos y desarrollo de aplicaciones.
- **Enfoque práctico y orientado a proyectos:** Facilita el aprendizaje a través de la implementación de proyectos reales y tangibles.
- **Orientación vocacional:** Ayuda a los estudiantes a confirmar su interés y aptitudes en el área tecnológica.
- **Preparación para desafíos futuros:** Proporciona una transición más suave y efectiva hacia la educación superior y el mundo laboral.

3.3. IMPACTO ESPERADO:

El programa **Juniority** espera no solo mejorar las competencias técnicas de los estudiantes, sino también fortalecer sus habilidades interpersonales y su confianza para enfrentar los desafíos académicos y profesionales. Al finalizar el curso, los estudiantes habrán adquirido las herramientas necesarias para destacar en cualquier entorno tecnológico, habrán realizado proyectos prácticos que enriquecerán su portafolio y, lo más importante, habrán cultivado una mentalidad orientada a la resolución de problemas y la innovación.

En resumen, este preuniversitario tecnológico no solo se centra en "enseñar a programar", sino en preparar a una nueva generación de estudiantes apasionados, resilientes y capaces de adaptarse a un mundo en constante transformación.

4. OBJETIVOS DEL PROYECTO

4.1. OBJETIVO GENERAL

Proporcionar a estudiantes de enseñanza media una formación integral en programación, bases de datos, desarrollo web y habilidades transversales, mediante un enfoque práctico y orientado a proyectos, que facilite su transición a estudios superiores o al ámbito laboral tecnológico.

4.2. OBJETIVOS ESPECÍFICOS

1. **Desarrollar habilidades en programación:**

- Introducir los fundamentos de programación mediante lenguajes accesibles y ampliamente utilizados como Java, JavaScript y SQL.
- Desarrollar la capacidad de resolver problemas a través de algoritmos y estructuras de datos.

2. **Enseñar diseño y prototipado de aplicaciones:**

- Capacitar a los estudiantes en el uso de herramientas de diseño visual como Figma y Miro.
- Guiar en la creación de prototipos funcionales para aplicaciones web y móviles.

3. **Implementar bases de datos eficientes:**

- Introducir conceptos esenciales de bases de datos relacionales.
- Facilitar la creación y manipulación de bases de datos mediante herramientas como MySQL Workbench y PostgreSQL.

4. **Desarrollar aplicaciones completas:**

- Integrar conocimientos de frontend (HTML, CSS, JavaScript) y backend (Node.js) para el desarrollo de aplicaciones web completas.
- Enseñar el uso de frameworks modernos como React para el desarrollo de interfaces interactivas.

5. **Fomentar habilidades transversales:**

- Promover el trabajo en equipo, la comunicación efectiva y la gestión de proyectos mediante metodologías ágiles.
- Desarrollar habilidades críticas como el pensamiento lógico y la resolución de problemas.

6. **Guiar el desarrollo de un proyecto final integral:**

- Acompañar a los estudiantes en el diseño, desarrollo y presentación de un proyecto tecnológico final.
- Crear una experiencia que simule un entorno de trabajo profesional.

7. **Brindar orientación vocacional:**

- Ofrecer sesiones informativas sobre las posibles carreras en el ámbito tecnológico.
- Proporcionar recursos y herramientas para la toma de decisiones académicas y profesionales informadas.

5. DESCRIPCIÓN GENERAL DEL PROGRAMA

El programa **Juniority: Preuniversitario Tecnológico** está diseñado para ofrecer una formación integral en habilidades tecnológicas clave, proporcionando a los estudiantes una experiencia educativa práctica, accesible y alineada con las demandas actuales del mercado, a través de una estructura modular, una metodología participativa y un enfoque pedagógico centrado en el estudiante.

5.1. METODOLOGÍA DE ENSEÑANZA

La metodología de enseñanza del programa Juniority se basa en los siguientes pilares:

- **Aprendizaje Basado en Proyectos (ABP):** Los estudiantes trabajarán en proyectos reales que les permitirán aplicar los conocimientos adquiridos en contextos prácticos. Cada unidad (semanal) culminará con un proyecto grupal específico que integrará los contenidos aprendidos.
- **Clases Teórico-Prácticas:** Cada clase combinará la exposición de conceptos teóricos con actividades prácticas inmediatas, permitiendo que los estudiantes consoliden sus conocimientos a través de ejercicios aplicados.
- **Trabajo Colaborativo:** Los estudiantes se organizarán en equipos para desarrollar actividades grupales, fomentando habilidades de comunicación, liderazgo y resolución de conflictos.
- **Sesiones Grupales y de Seguimiento Personalizado:** El programa incluye sesiones grupales para la entrega y explicación de contenidos, y sesiones personalizadas donde el docente actúa como un guía o "director de proyecto", apoyando a cada grupo en su avance.
- **Retroalimentación Continua:** Los docentes proporcionarán retroalimentación constante a través de correcciones formativas, fomentando la mejora continua de los estudiantes.

- **Uso de Herramientas Digitales:** Plataformas como Figma, Miro, MySQL Workbench, PostgreSQL, NetBeans, Visual Studio Code y entornos de desarrollo integrados (IDEs) serán herramientas esenciales durante el curso, garantizando que los estudiantes adquieran experiencia en herramientas ampliamente utilizadas en la industria.

5.2. DURACIÓN Y ESTRUCTURA TEMPORAL

El programa tiene una duración total de 8 meses, divididos en 8 módulos consecutivos, cada uno enfocado en un área específica del desarrollo tecnológico.

Estructura General del Programa:

- **Duración total:** 8 meses.
- **Frecuencia:** 2 sesiones semanales (1 sesión grupal de 2 horas + 1 sesión por equipo de 1 hora).
- **Carga Horaria Total:** Aproximadamente 160 horas lectivas.

Distribución Temporal de los Módulos:

- **Prototipos y Diseño (Figma, Miro, MySQL Workbench):** 1 mes.
- **Fundamentos de SQL y bases de datos:** 1 mes.
- **Programación con Java y conexión a bases de datos:** 1 mes.
- **Desarrollo web con HTML, CSS y JavaScript:** 1 mes.
- **React y desarrollo frontend:** 1 mes.
- **Backend con Node.js y APIs REST:** 1 mes.
- **PostgreSQL y consultas avanzadas:** 1 mes.
- **Proyecto Final:** 1 mes.

Cada módulo está diseñado para garantizar un aprendizaje progresivo, con entregables semanales y una evaluación al final de cada etapa.

5.3. ENFOQUE PEDAGÓGICO

El enfoque pedagógico del programa **Juniority** está centrado en el estudiante, reconociendo sus necesidades individuales y su estilo de aprendizaje. Se sustenta en los siguientes principios:

- **Aprendizaje Activo:** Los estudiantes son protagonistas de su propio proceso de aprendizaje, participando activamente en discusiones, actividades prácticas y proyectos colaborativos.
- **Personalización del Aprendizaje:** Se adaptarán los contenidos y ritmos de aprendizaje a las capacidades individuales de cada estudiante, asegurando que todos puedan avanzar de manera efectiva.
- **Desarrollo de Competencias Transversales:** Además de habilidades técnicas, el programa fortalecerá competencias como la comunicación efectiva, el pensamiento crítico, la gestión del tiempo y el trabajo en equipo.
- **Evaluación Formativa y Sumativa:** A lo largo del curso, se utilizarán herramientas de evaluación tanto formativas (retroalimentación continua) como sumativas (proyectos finales y pruebas prácticas).
- **Integración de Herramientas Digitales:** Se promoverá el uso de herramientas digitales actuales, preparando a los estudiantes para entornos profesionales reales.
- **Acompañamiento y Mentoría:** Los docentes no solo enseñarán, sino que también actuarán como mentores, guiando y motivando a los estudiantes en su proceso de aprendizaje.

6. MÓDULOS DEL PROGRAMA

6.1. PROTOTIPOS Y DISEÑO

- **Objetivos:**
 - Comprender el proceso de diseño de aplicaciones desde la idea inicial hasta un prototipo funcional.
 - Aprender a usar herramientas como Figma para diseñar interfaces de usuario (UI) atractivas y funcionales.
 - Diseñar un modelo de datos básico utilizando **Workbench** o herramientas similares.
 - Entender la relación entre el diseño de la interfaz y la estructura de datos subyacente.

6.2. FUNDAMENTOS DE SQL Y BASES DE DATOS

- **Objetivos:**
 - Comprender los conceptos básicos de bases de datos relacionales y su importancia en el desarrollo de aplicaciones.
 - Aprender a crear y estructurar bases de datos utilizando herramientas como MySQL.
 - Escribir consultas SQL para insertar, actualizar, eliminar y extraer datos.
 - Diseñar tablas y relaciones básicas siguiendo un modelo previamente creado.

6.3. PROGRAMACIÓN ORIENTADA A OBJETOS

- **Objetivos:**
 - Aprender los fundamentos de programación en Java, incluyendo variables, clases, polimorfismo y estructuras de datos.
 - Escribir algoritmos básicos y resolver problemas prácticos utilizando Java.
 - Conectar aplicaciones Java a bases de datos utilizando JDBC.
 - Diseñar una pequeña aplicación de consola que interactúe con una base de datos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
 - Implementar buenas prácticas en programación orientada a objetos.

6.4. DESARROLLO WEB CON HTML, CSS Y JAVASCRIPT

- **Objetivos:**
 - Comprender la estructura básica de una página web utilizando HTML.
 - Aplicar estilos visuales a páginas web utilizando CSS, incluyendo diseño responsivo.
 - Introducir interactividad básica en páginas web mediante JavaScript.
 - Implementar manipulación del DOM para responder a eventos del usuario (clics, entradas, etc.).
 - Crear una página web funcional con diseño y comportamiento interactivo.

6.5. REACT Y DESARROLLO FRONTEND

- **Objetivos:**
 - Comprender los fundamentos de React, incluyendo componentes, estados y props.
 - Construir interfaces dinámicas y reutilizables utilizando React.
 - Manejar eventos y formularios dentro de una aplicación React.
 - Integrar APIs externas para mostrar datos dinámicos en la aplicación.
 - Diseñar y desarrollar un frontend funcional para una aplicación basada en React y respaldada mediante Git.

6.6. BACKEND CON NODE.JS Y APIS REST

- **Objetivos:**
 - Comprender el rol de un backend en una aplicación web.
 - Escribir servidores básicos utilizando Node.js y Express.js.
 - Diseñar y desarrollar APIs RESTful para manejar datos y solicitudes del cliente.
 - Manejar solicitudes HTTP (GET, POST, PUT, DELETE) y devolver respuestas adecuadas.
 - Implementar middleware básico para gestionar errores y validar datos.

6.7. POSTGRESQL Y CONSULTAS AVANZADAS

- **Objetivos:**
 - Aprender las características avanzadas de PostgreSQL y sus diferencias con otras bases de datos.
 - Diseñar bases de datos relacionales más complejas con varias relaciones y restricciones.
 - Escribir consultas avanzadas utilizando joins, subconsultas y funciones agregadas.
 - Implementar diferentes filtros de búsqueda, paginación y validaciones (JWT)
 - Integrar PostgreSQL con aplicaciones de React.

6.8. PROYECTO FINAL

- **Objetivos:**
 - Diseñar, desarrollar y presentar una aplicación web funcional que integre frontend, backend y base de datos.
 - Aplicar conocimientos de diseño (Figma), desarrollo web (HTML, CSS, JavaScript, React) y backend (Node.js, PostgreSQL).
 - Trabajar en un entorno similar al desarrollo profesional, incluyendo la planificación, división de tareas, gestión de entregables y respaldo de proyecto en repositorio (GitHub).
 - Documentar el proyecto, explicando su arquitectura, decisiones de diseño y futuras mejoras (ERS).
 - Presentar el proyecto ante un público, destacando las funcionalidades clave y los retos superados.

7. PERFIL DEL ESTUDIANTE EGRESADO

Al finalizar el programa **Juniority: Preuniversitario Tecnológico**, los estudiantes habrán adquirido un conjunto integral de habilidades técnicas y blandas que les permitirán enfrentar con confianza los desafíos del mundo académico y profesional en el ámbito tecnológico.

7.1 HABILIDADES TÉCNICAS

Al finalizar el programa Juniority: Preuniversitario Tecnológico, los estudiantes habrán adquirido un conjunto integral de habilidades técnicas y blandas que les permitirán enfrentar con confianza los desafíos del mundo académico y profesional en el ámbito tecnológico.

- **Lenguajes de Programación:** Fundamentos sólidos en Java, JavaScript, SQL y principios de programación orientada a objetos.
- **Bases de Datos:** Diseño, implementación y manipulación de bases de datos utilizando MySQL y PostgreSQL.
- **Desarrollo Frontend:** Creación de interfaces dinámicas y atractivas con HTML, CSS y React.

- **Desarrollo Backend:** Interacción con base de datos mediante JBDC utilizando NetBeans y Java. Construcción de servidores y APIs REST utilizando Node.js y Express.js.
- **Herramientas de Diseño:** Prototipado y diseño de interfaces con Figma, Canva y Miro.
- **Modelado de Datos:** Creación de diagramas entidad-relación (ER) y esquemas funcionales para bases de datos robustas.

Estas habilidades técnicas garantizarán que los estudiantes puedan enfrentar proyectos prácticos en entornos reales, siguiendo estándares y buenas prácticas de la industria tecnológica.

7.2 HABILIDADES BLANDAS

El programa no solo se centra en las habilidades técnicas, sino que también fortalece competencias transversales esenciales para el éxito en cualquier entorno profesional:

- **Trabajo en Equipo:** Capacidad para colaborar efectivamente, comunicar ideas y contribuir al logro de objetivos comunes.
- **Pensamiento Crítico:** Habilidad para analizar problemas complejos, evaluar soluciones y tomar decisiones informadas.
- **Resolución de Problemas:** Enfoque proactivo para identificar y abordar desafíos con creatividad y lógica.
- **Adaptabilidad:** Capacidad para aprender nuevas herramientas y tecnologías rápidamente.
- **Gestión del Tiempo:** Organización y planificación efectiva para cumplir plazos y objetivos.

7.3 PROYECCIÓN PROFESIONAL

El perfil del egresado de Juniority está diseñado para abrir diversas oportunidades, tanto en el ámbito académico como en el laboral:

- **Continuación de Estudios:** Los estudiantes estarán preparados para ingresar a carreras universitarias o técnicas en áreas como Ingeniería en Informática, Desarrollo de Software, Analista Programador Computacional, Diseño UX/UI, entre otras.
- **Empleabilidad Temprana:** Gracias a las habilidades adquiridas, los estudiantes podrán desempeñarse en roles de nivel junior como:
 - Asistente de Desarrollo Web
 - Soporte Técnico en Bases de Datos
 - Desarrollador Junior Frontend o Backend
 - Asistente en Diseño UX/UI
- **Emprendimiento Tecnológico:** Los estudiantes tendrán las bases para iniciar sus propios proyectos tecnológicos o colaborar en startups y pequeñas empresas.

8. METODOLOGÍA DE EVALUACIÓN

El programa **Juniority: Preuniversitario Tecnológico** adopta una metodología de evaluación equilibrada que combina la retroalimentación continua con mediciones objetivas del desempeño académico. El sistema está diseñado para asegurar que los estudiantes adquieran no solo conocimientos teóricos, sino también habilidades prácticas y una mentalidad reflexiva frente a sus propios procesos de aprendizaje.

8.1 EVALUACIONES SEMANALES: ENFOQUE FORMATIVO

Las **evaluaciones semanales** tienen un carácter **formativo** y están orientadas a monitorear el progreso de los estudiantes de manera continua. Estas instancias permiten al docente:

- **Identificar dificultades específicas** en el aprendizaje de los estudiantes.
- **Proporcionar retroalimentación oportuna y personalizada** para corregir errores y reforzar conceptos.
- **Fomentar la autorreflexión** en los estudiantes sobre su propio desempeño.

Características de las evaluaciones semanales:

- Actividades prácticas: ejercicios individuales y grupales.
- Entregas de tareas asignadas en cada clase.
- Participación activa en las sesiones prácticas.
- Evaluación de ejercicios realizados en herramientas específicas.

Estas evaluaciones no tienen una calificación numérica definitiva, pero sí son un requisito para la participación en las evaluaciones mensuales.

8.2 EVALUACIONES MENSUALES: ENFOQUE SUMATIVO

Al finalizar cada **módulo mensual**, los estudiantes deberán rendir una **evaluación sumativa** que medirá de forma objetiva los conocimientos adquiridos y su aplicación práctica. Estas evaluaciones están diseñadas para validar el progreso del estudiante respecto a los objetivos específicos de cada módulo.

Características de las evaluaciones mensuales:

- Exámenes prácticos: resolución de problemas reales utilizando las herramientas enseñadas.
- Proyectos parciales: desarrollo de tareas específicas dentro de un contexto práctico.
- Evaluación individual y grupal: análisis del desempeño tanto personal como colaborativo.

Las calificaciones obtenidas en estas evaluaciones representan el 60% de la nota final del curso.

8.3 PROYECTO FINAL: INTEGRACIÓN DE APRENDIZAJES

El **proyecto final** es la instancia más importante del programa, donde los estudiantes deberán aplicar todos los conocimientos y habilidades adquiridas a lo largo del curso. Este proyecto será evaluado bajo los siguientes criterios:

- **Aplicación práctica de los conceptos aprendidos:** Uso adecuado de lenguajes de programación, bases de datos, diseño de interfaces y desarrollo.

- **Trabajo en equipo:** Colaboración efectiva y roles bien definidos dentro del grupo.
- **Resolución de problemas:** Capacidad para identificar desafíos y proponer soluciones funcionales.
- **Presentación final:** Claridad y estructura en la entrega y exposición del proyecto.

El proyecto final representa el **40% de la nota final del curso.**

9. RECURSOS EDUCATIVOS Y HERRAMIENTAS

En el programa Juniority: Preuniversitario Tecnológico, se utilizará una combinación de plataformas digitales, materiales de apoyo y requisitos técnicos cuidadosamente seleccionados para optimizar el aprendizaje y permitir que los estudiantes desarrollen habilidades prácticas aplicables en entornos reales.

9.1 PLATAFORMAS DIGITALES

Las siguientes herramientas digitales serán esenciales para las actividades prácticas y el desarrollo de proyectos dentro del curso:

- **Figma:** Plataforma de diseño colaborativo para crear prototipos y diseños de interfaces de usuario (UI). Permite a los estudiantes visualizar y planificar estructuras de aplicaciones de forma intuitiva.
- **Miro:** Plataforma de colaboración visual en línea diseñada para la creación de diagramas, mapas conceptuales y flujos de trabajo. Permite a los estudiantes y equipos planificar, organizar y visualizar ideas de manera clara y colaborativa. Es especialmente útil para crear diagramas de flujo, mapas de navegación y esquemas de planificación de proyectos.
- **MySQL Workbench:** Herramienta de modelado y gestión de bases de datos relacionales. Los estudiantes aprenderán a diseñar esquemas de datos, crear tablas y establecer relaciones entre ellas.

- **PostgreSQL:** Sistema de gestión de bases de datos relacional de código abierto, conocido por su robustez, flexibilidad y cumplimiento de estándares SQL. PostgreSQL permite almacenar, manipular y consultar grandes volúmenes de datos de manera eficiente. Es ideal para aplicaciones que requieren integridad de datos y escalabilidad, y es ampliamente utilizado en entornos empresariales y académicos.
- **Visual Studio Code:** Editor de código fuente versátil y potente, ideal para escribir y depurar código en múltiples lenguajes de programación como JavaScript y SQL.
- **NetBeans IDE:** Entorno de desarrollo integrado (IDE) diseñado para programar en Java. Facilitará la escritura, compilación y depuración de aplicaciones Java, especialmente aquellas conectadas a bases de datos.
- **Node.js:** Entorno de tiempo de ejecución para JavaScript que permite el desarrollo de aplicaciones backend. Se utilizará para construir servidores y APIs REST.
- **Zoom:** Plataforma de videoconferencias para clases en vivo. Permite realizar clases virtuales en tiempo real con docentes y compañeros facilitando la interacción directa y resolución de dudas al instante.
- **LMS (Learning Management System) de Juniority:** Plataforma de aprendizaje virtual diseñada exclusivamente para el programa Juniority. Permite a los estudiantes acceder a materiales educativos, descargar recursos del curso, enviar tareas y proyectos, participar en foros de discusión y realizar un seguimiento detallado de su progreso académico.
- **Canva:** Plataforma de diseño gráfico en línea que permite crear presentaciones, prototipos y materiales visuales de forma intuitiva y colaborativa. Facilita la creación de recursos educativos atractivos, como infografías, diapositivas y contenido multimedia, optimizando la comunicación visual dentro del programa Juniority.

9.2 MATERIAL DE APOYO

El programa contará con una variedad de recursos didácticos diseñados para complementar las clases prácticas y guías de estudio:

- **Computador personal (PC o Laptop):** Procesador de al menos **Intel i3** o equivalente.
- **Sistema Operativo:** Windows 10 o superior, macOS, o distribuciones de Linux compatibles.

- **Conexión a Internet estable:** Velocidad mínima de **10 Mbps** para videollamadas y descargas de software.
- **Software necesario preinstalado:** MySQL Workbench, NetBeans IDE, Visual Studio Code y Node.js.
- **Cámara y micrófono funcionales:** Para participar en sesiones virtuales y reuniones grupales.
- **Cuenta en plataformas educativas:** Acceso a plataformas como GitHub, Google Drive y otros servicios utilizados en el curso.

El correcto uso de estas herramientas y materiales permitirá que los estudiantes:

- Desarrollen proyectos funcionales desde el diseño hasta su implementación.
- Trabajen de manera colaborativa en un entorno que simula prácticas profesionales.
- Optimicen sus flujos de trabajo mediante herramientas de última generación.
- Aprendan a enfrentar problemas reales utilizando tecnología estándar en la industria.

10. ROL DEL DOCENTE

El rol del docente en el programa **Juniority** es fundamental para garantizar un proceso de enseñanza-aprendizaje efectivo, inclusivo y alineado con los objetivos establecidos. Más allá de transmitir conocimientos técnicos, el docente actúa como guía, mentor y facilitador, creando un entorno donde los estudiantes puedan desarrollar tanto habilidades técnicas como blandas.

10.1 FUNCIONES PRINCIPALES:

- **Facilitador:** Proporciona los recursos, materiales y herramientas necesarias para que los estudiantes puedan adquirir conocimientos de forma autónoma y práctica.

- **Mentor:** Acompaña a los estudiantes en su desarrollo profesional y personal, ofreciendo orientación para superar desafíos técnicos y emocionales.
- **Guía:** Dirige el proceso de aprendizaje, asegurando que los estudiantes avancen según el plan curricular y cumplan con los objetivos establecidos en cada módulo.
- **Evaluador:** Diseña y aplica instrumentos de evaluación formativa y sumativa, proporcionando retroalimentación constructiva para fomentar la mejora continua.

10.2 INTERACCIÓN CON LOS GRUPOS:

El docente interactúa con los estudiantes en dos modalidades clave:

- **Sesiones Grupales:** Clases teóricas y prácticas donde se abordan los contenidos del programa, se desarrollan ejercicios y se resuelven dudas de manera colectiva.
- **Sesiones Individuales o por Grupo de Trabajo:** Espacios personalizados para guiar proyectos, resolver problemas específicos y fortalecer las habilidades individuales de cada estudiante. Podrán ser conducidas por el docente o un tutor.

En ambas modalidades, el docente fomenta un ambiente de participación activa, respeto mutuo y comunicación abierta.

10.3 SEGUIMIENTO DEL PROGRESO:

El progreso de los estudiantes es monitoreado de manera constante a través de:

- **Reportes Periódicos:** Evaluaciones de desempeño en ejercicios prácticos, participación en clase y cumplimiento de objetivos semanales.
- **Reuniones de Seguimiento:** Sesiones individuales o grupales donde se revisa el avance, se resuelven bloqueos y se ajustan estrategias para alcanzar los objetivos propuestos.
- **Retroalimentación Constructiva:** Comentarios claros y específicos que permiten a los estudiantes identificar fortalezas y áreas de mejora.

El docente, como figura central en el proceso de aprendizaje, no solo guía técnicamente a los estudiantes, sino que también fomenta una cultura de colaboración, confianza y autonomía.

11. EVALUACIÓN Y SEGUIMIENTO DEL PROYECTO

El éxito del programa **Juniority** no solo depende de la entrega de contenidos y la participación activa de los estudiantes, sino también de un sistema de evaluación y seguimiento continuo que permita medir los avances, detectar áreas de mejora y asegurar que los objetivos propuestos se estén cumpliendo.

11.1 INDICADORES CLAVE DE ÉXITO (KPIs)

Los Indicadores Clave de Éxito (KPIs) son métricas utilizadas para evaluar el progreso y los resultados del programa. Estos permiten obtener una visión objetiva del desempeño de los estudiantes y la efectividad de las estrategias implementadas.

- **Asistencia:** El compromiso de los estudiantes con el programa se mide a través de su presencia y participación activa en las sesiones grupales e individuales.
- **Entrega de Proyectos:** La finalización y entrega de los proyectos semanales y mensuales son esenciales para evaluar el aprendizaje práctico y la aplicación de los conocimientos adquiridos.
- **Satisfacción del Estudiante:** Se medirá a través de encuestas periódicas que capturen la percepción de los estudiantes sobre el contenido, la dinámica de las clases y el apoyo recibido.

11.2 RETROALIMENTACIÓN DE ESTUDIANTES Y APODERADOS

La retroalimentación es una herramienta fundamental para el perfeccionamiento continuo del programa. Se implementarán las siguientes estrategias:

- **Encuestas Periódicas a Estudiantes:** Al finalizar cada módulo, se aplicarán encuestas para conocer su experiencia, detectar desafíos y obtener sugerencias para mejorar el curso.

- **Encuestas a Apoderados (cuando corresponda):** Los apoderados podrán compartir su percepción sobre el avance de sus hijos/as y el impacto del programa en su formación.
- **Sesiones de Retroalimentación Directa:** Espacios donde los estudiantes puedan expresar libremente sus inquietudes y sugerencias al docente y al equipo coordinador.

11.3 INFORMES DE AVANCE

Para garantizar un seguimiento detallado y sistemático, se implementarán informes regulares que reflejen los logros alcanzados y los desafíos pendientes:

- **Informes Mensuales:** Resumen de la asistencia, participación, entregas y desempeño académico de cada estudiante.
- **Informes Trimestrales:** Evaluación más profunda sobre el progreso del grupo, desempeño individual y recomendaciones para optimizar el proceso de aprendizaje.
- **Reuniones de Seguimiento con el Equipo Docente:** Espacios donde el equipo pedagógico revisa los informes y ajusta las estrategias para enfrentar desafíos detectados.

12. PLAN DE COMUNICACIÓN

La comunicación efectiva es un pilar fundamental para el éxito del programa **Juniority**. Un flujo claro y constante de información entre docentes, estudiantes y apoderados asegura que todos los participantes estén alineados con los objetivos, expectativas y avances del curso. A continuación, se describen los principales elementos que conforman el plan de comunicación.

12.1 COMUNICACIÓN CON ALUMNOS Y APODERADOS

El contacto directo y oportuno con los estudiantes y sus familias es esencial para resolver dudas, reforzar compromisos y mantener un ambiente de confianza y colaboración.

- **Canales Oficiales de Comunicación:**

- **Correo Electrónico:** Para comunicaciones formales, envío de informes y recordatorios importantes.
- **Plataforma Educativa:** Espacio principal para compartir materiales de clase, tareas, fechas importantes y recursos adicionales.
- **Mensajería Instantánea (Slack/Discord):** Grupos moderados por el docente para resolver dudas puntuales y brindar recordatorios rápidos.

- **Reuniones con Apoderados:**

- Conversaciones individuales para abordar casos o desafíos específicos.

12.2 ATENCIÓN DE DUDAS Y CONSULTAS

Contar con canales claros para resolver dudas promueve la confianza y asegura que los estudiantes puedan avanzar sin obstáculos en su aprendizaje.

- **Horarios de Atención:**

- Horas específicas durante la semana en las que el docente estará disponible para resolver dudas.
- Atención individualizada bajo solicitud previa.

- **Medios Disponibles:**

- Plataforma Educativa: Para consultas académicas y materiales de referencia.
- Correo Electrónico: Para dudas más extensas o formales.
- Sesiones Grupales: Espacios abiertos para resolver inquietudes comunes.

12.3 PROTOCOLO DE COMUNICACIÓN

- Se establecerán normas claras para el uso adecuado de los canales oficiales.
- Se promoverá un lenguaje respetuoso y profesional en todas las interacciones.
- Los tiempos de respuesta estarán definidos para cada canal (ej., correos respondidos en un máximo de 48 horas hábiles).

13. CIERRE DEL PROGRAMA Y CERTIFICACIÓN

El cierre del programa **Juniority** es una etapa fundamental que celebra los logros alcanzados por los estudiantes, valida su esfuerzo y refuerza su confianza para enfrentar futuros desafíos académicos y profesionales. Este momento no solo marca el final del curso, sino también el inicio de nuevas oportunidades.

13.1 EVENTO DE CLAUSURA

El programa culminará con un evento especial donde los estudiantes tendrán la oportunidad de mostrar sus proyectos finales frente a sus compañeros, docentes, apoderados e invitados especiales.

- **Presentación de Proyectos Finales:**

- Cada grupo o estudiante presentará su proyecto final, explicando su funcionalidad, los desafíos superados y las lecciones aprendidas.
- Se fomentará un ambiente de colaboración y reconocimiento mutuo entre los participantes.
- Se podrán realizar demostraciones prácticas de las aplicaciones desarrolladas.

- **Espacio de Reflexión:**

- Los estudiantes compartirán sus experiencias durante el programa.
- Se destacará la importancia de la perseverancia, el trabajo en equipo y la resolución de problemas.

13.2 ENTREGA DE CERTIFICADOS POR MÓDULO

Al finalizar cada módulo, los estudiantes recibirán un **Certificado de Finalización de Módulo**, validando las competencias adquiridas en esa fase específica.

- **Características de los Certificados por Módulo:**

- Nombre completo del estudiante.
- Nombre del módulo completado (ej., Prototipos y Diseño, Fundamentos de SQL y Bases de Datos).
- Habilidades específicas adquiridas en el módulo.

- Horas lectivas cursadas.
- Firma del docente responsable y la dirección académica.
- Fecha de finalización del módulo.
- **Importancia de los Certificados por Módulo:**
 - Permiten que los estudiantes validen su avance de forma progresiva.
 - Brindan motivación al reconocer logros en cada etapa.
 - Facilitan la inclusión en portafolios académicos o profesionales.

13.3 CERTIFICADO FINAL DEL PROGRAMA

Al finalizar todos los módulos y aprobar el proyecto final, los estudiantes recibirán un Certificado Final de Egreso, el cual validará de manera integral todas las competencias adquiridas a lo largo del curso.

- **Título del Certificado Final:**
 - **"Programador Junior Full Stack"**.
- **Características del Certificado Final:**
 - Nombre completo del estudiante.
 - Título obtenido: **Programador Junior Full Stack.**
 - Listado de módulos completados con éxito.
 - Firma del docente principal, coordinación académica y dirección del programa.
 - Fecha de finalización.
 - Código único de validación.
- **Importancia del Certificado Final:**
 - Representa una validación formal de todas las competencias adquiridas.
 - Facilita la inclusión en currículums profesionales.
 - Puede servir como puente para acceder a otros programas de formación avanzada o prácticas profesionales.

13.4 RECONOCIMIENTOS ESPECIALES

Se otorgarán menciones especiales para destacar los esfuerzos y logros más sobresalientes durante el curso.

- **Categorías de Reconocimiento:**

- Mejor Desempeño Académico: Al estudiante con la mayor calificación acumulada.
- Mejor Proyecto Final: Al grupo o estudiante que presente el proyecto más innovador y funcional.
- Espíritu de Equipo: Al estudiante que haya demostrado un fuerte compromiso con la colaboración y apoyo a sus compañeros.

- **Entrega de Premios:**

- Los reconocimientos podrán incluir diplomas adicionales, medallas simbólicas o pequeñas recompensas materiales.

El evento de clausura no solo marca el cierre de una etapa, sino que también inspira a los estudiantes a continuar su camino en el mundo de la tecnología y la programación. Se destacará la importancia de aplicar las habilidades adquiridas en nuevos proyectos y desafíos.

El programa Juniority no termina con la entrega de un certificado, sino que deja una base sólida sobre la cual los estudiantes podrán seguir construyendo su futuro académico y profesional.

14. ANEXOS Y CONTACTOS

14.1 CONTACTO DEL EQUIPO

- **Correo Institucional:** contacto@juniority.cl
- **Sitio Web Oficial:** www.juniority.cl
- **Teléfono de Contacto:** +56 9 34615794

14.2 EQUIPO DIRECTIVO Y COORDINACIÓN

- **Coordinador del Programa:** Carlo Aguirre Chacón – info@juniority.cl

14.3 REDES SOCIALES OFICIALES

- **Instagram:** [@juniority_oficial](https://www.instagram.com/juniority_oficial)
- **Facebook:** [facebook.com/JuniorityOficial](https://www.facebook.com/JuniorityOficial)
- **LinkedIn:** [linkedin.com/company/juniority_oficial](https://www.linkedin.com/company/juniority_oficial)